

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

*Кафедра автоматизованих систем обробки інформації і управління*

До захисту допущено:

В.о. завідувача кафедри

\_\_\_\_\_ Олександр ПАВЛОВ  
(підпис) (вл.ім'я, прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2020 р.

**Дипломний проєкт**  
на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інформаційні управляючі  
системи та технології»  
спеціальності 122 «Комп'ютерні науки та інформаційні технології»

на тему: «Інформаційна технологія визначення тональності  
текстів»

---

**Виконав :** студент IV курсу, групи ІС-63

\_\_\_\_\_ Блінков Євген Миколайович  
(прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

**Керівник**

\_\_\_\_\_ доц., к.т.н., доц. Жданова Олена Григорівна  
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

**Консультант з  
графічної  
документації**

\_\_\_\_\_ ст. вик. Проскура Світлана Леонідівна  
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

**Рецензент**

\_\_\_\_\_ проф., д.т.н., доц. Клименко Ірина Анатоліївна  
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

Засвідчую, що у цьому дипломному проєкті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2020 року

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки  
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління  
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ  
(підпис) (вл.ім'я, прізвище)

“ \_\_\_ ” \_\_\_\_\_ 2020 р.

**ЗАВДАННЯ  
на дипломний проєкт студенту**

Блінкову Євгену Миколайовичу  
(прізвище, ім'я, по батькові)

1. Тема проєкту «Інформаційна технологія визначення тональності текстів»

керівник проєкту Жданова Олена Григорівна, к.т.н., доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “07” травня 2020 р. №1081-с

2. Термін подання студентом проєкту “01” червня 2020 року

3. Вихідні дані до проєкту

*Технічне завдання*

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1. *Схема структурна варіантів використань*
2. *Схема структурна станів системи*
3. *Схема бази даних*
4. *Схема структурна класів програмного забезпечення*
5. *Схема структурна послідовності*
6. *Рішення з математичного забезпечення*
7. *Креслення вигляду екранних форм*

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «13» квітня 2020 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	<i>Вивчення рекомендованої літератури</i>	<i>17.04.2020</i>	
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>18.04.2020</i>	
3.	<i>Постановка та формалізація задачі</i>	<i>20.04.2020</i>	
4.	<i>Розробка інформаційного забезпечення</i>	<i>22.04.2020</i>	
5.	<i>Алгоритмізація задачі</i>	<i>25.04.2020</i>	
6.	<i>Обґрунтування використовуваних технічних засобів</i>	<i>26.04.2020</i>	
7.	<i>Розробка програмного забезпечення</i>	<i>01.05.2020</i>	
8.	<i>Налагодження програми</i>	<i>05.05.2020</i>	
9.	<i>Виконання графічних документів</i>	<i>10.05.2020</i>	
10.	<i>Оформлення пояснювальної записки</i>	<i>13.05.2020</i>	
11.	<i>Подання ДП на попередній захист</i>	<i>14.05.2020</i>	
12.	<i>Подання ДП на основний захист</i>	<i>01.06.2020</i>	
13.	<i>Подання ДП рецензенту</i>	<i>02.06.2020</i>	

Студент

Євген БЛІНКОВ

Керівник

Олена ЖДАНОВА



## АНОТАЦІЯ

**Структура та обсяг роботи.** Пояснювальна записка дипломного проекту складається з п'яти розділів, містить 26 рисунків, 7 таблиць, 1 додаток, 17 джерел.

Дипломний проект присвячений автоматизації процесів проведення аналізу тональності текстів, застосовуючи різні алгоритми та порівняння ефективності цих алгоритмів. В даному проекті описуються методи визначення тональності текстів та способи їх застосування при розробці інформаційної технології визначення тональності текстів.

У розділі інформаційного забезпечення були наведені набори вхідних та вихідних даних, а також описано їх формат, структуру та призначення у програмному продукті.

Розділ математичного забезпечення, насамперед, присвячений опису змістовної та математичної постановок задачі, а також ключових методів розв'язання поставленої задачі. Крім цього, наводиться обґрунтування вибору даних методів для їх реалізації у програмному продукті.

У розділі програмного забезпечення наводяться засоби розробки, які були використані у даному програмному продукті, а також описується принцип роботи веб-застосування у вигляді різних діаграм.

У технологічному розділі наводиться керівництво користувачу при користуванні даною програмою, а також описуються результати випробувань.

**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ, ВИЗНАЧЕННЯ  
ТОНАЛЬНОСТІ ТЕКСТУ, НАЇВНИЙ БАЄСІВ КЛАСИФІКАТОР,  
НЕЙРОННА МЕРЕЖА, АВТОМАТИЗАЦІЯ.**

					<b>ДП 6303.00.000 ПЗ</b>			
<i>Зм.</i>	<i>Арк.</i>	<i>Прізвище</i>	<i>Підпис</i>	<i>Дата</i>	<i>Інформаційна технологія визначення тональності текстів</i>	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
<i>Розроб.</i>		<i>Блінков Є.М.</i>						
<i>Перевірив.</i>		<i>Жданова О.Г.</i>					2	83
<i>Н. кон.</i>		<i>Проскура С.Л.</i>				<i>КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63</i>		
<i>Зав.</i>		<i>Павлов О.А.</i>						

## ABSTRACT

**Structure and scope of work.** The explanatory note of the diploma project consists of five sections, contains 26 figures, 7 tables, 1 appendix, 17 sources.

The diploma project is devoted to automation of processes of the sentimental analysis of texts, applying various algorithms, and comparison of efficiency of these algorithms. This project describes the methods of sentimental analysis of texts and principles of their application in the development of information technology for text sentimental analysis.

The information support section provided sets of input and output data, as well as described their format, structure and purpose in the software product.

The section of mathematical support is primarily devoted to the description of meaningful and mathematical formulations of the problem, as well as key methods for solving the problem. In addition, the rationale for the choice of these methods for their implementation in the software product.

The software section lists the development tools that have been used in this software product, as well as how the web application works in the form of various diagrams.

The technological section provides user guidance for using this program, as well as describes the test results.

INFORMATION TECHNOLOGY, SENTIMENTAL ANALYSIS  
 NAÏVE BAYES CLASSIFIER, NEURAL NETWORK, AUTOMATION.

					ДП 6303.00.000 ПЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

## ЗМІСТ

ВСТУП .....	6
<b>1 ЗАГАЛЬНІ ПОЛОЖЕННЯ .....</b>	<b>8</b>
<b>1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА .....</b>	<b>8</b>
<i>1.1.1 Опис процесу діяльності .....</i>	<i>9</i>
<i>1.1.2 Опис функціональної моделі .....</i>	<i>12</i>
<b>1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ .....</b>	<b>14</b>
<b>1.3 ПОСТАНОВКА ЗАДАЧІ .....</b>	<b>16</b>
<i>1.3.1 Призначення розробки .....</i>	<i>16</i>
<i>1.3.2 Цілі та задачі розробки .....</i>	<i>16</i>
<b>ВИСНОВОК ДО РОЗДІЛУ .....</b>	<b>17</b>
<b>2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>18</b>
<b>2.1 ВХІДНІ ДАНІ .....</b>	<b>18</b>
<b>2.2 ВИХІДНІ ДАНІ .....</b>	<b>18</b>
<i>2.2.1 Перелік вихідних сигналів .....</i>	<i>18</i>
<i>2.2.2 Перелік вихідних документів .....</i>	<i>19</i>
<b>2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ .....</b>	<b>19</b>
<b>2.4 СТРУКТУРА МАСИВІВ ІНФОРМАЦІЇ .....</b>	<b>20</b>
<b>ВИСНОВОК ДО РОЗДІЛУ .....</b>	<b>26</b>
<b>3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>27</b>
<b>3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ .....</b>	<b>27</b>
<b>3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ .....</b>	<b>28</b>
<b>3.3 ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ .....</b>	<b>29</b>
<b>3.4 ОПИС МЕТОДІВ РОЗВ'ЯЗАННЯ .....</b>	<b>30</b>
<i>3.4.1 Найвіший баєсів класифікатор .....</i>	<i>30</i>
<i>3.4.2 Рекурентна нейронна мережа з ДКЧП .....</i>	<i>32</i>
<i>3.4.3 Згорткова нейронна мережа .....</i>	<i>34</i>
<i>3.4.4 Комбінація згорткової та рекурентної нейронної мережі .....</i>	<i>35</i>
<b>ВИСНОВОК ДО РОЗДІЛУ .....</b>	<b>36</b>

<b>4</b>	<b>ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>37</b>
4.1	ЗАСОБИ РОЗРОБКИ.....	37
4.2	ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ .....	38
4.2.1	<i>Загальні вимоги .....</i>	38
4.3	АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	38
4.3.1	<i>Діаграма класів.....</i>	38
4.3.2	<i>Діаграма послідовності.....</i>	39
4.3.3	<i>Діаграма компонентів.....</i>	40
4.3.4	<i>Специфікація функцій .....</i>	41
4.4	ОПИС ЗВІТІВ.....	44
	ВИСНОВОК ДО РОЗДІЛУ .....	46
<b>5</b>	<b>ТЕХНОЛОГІЧНИЙ РОЗДІЛ .....</b>	<b>47</b>
5.1	КЕРІВНИЦТВО КОРИСТУВАЧА .....	47
5.2	ВИПРОБУВАННЯ ПРОГРАМНОГО ПРОДУКТУ .....	54
5.2.1	<i>Мета випробувань.....</i>	54
5.2.2	<i>Загальні положення.....</i>	54
5.2.3	<i>Результати випробувань.....</i>	54
	ВИСНОВОК ДО РОЗДІЛУ .....	61
	ЗАГАЛЬНІ ВИСНОВКИ .....	62
	ПЕРЕЛІК ПОСИЛАНЬ .....	64
	ДОДАТОК А.....	66

## ВСТУП

За останнє десятиліття загальний обсяг інформації, яку слід обробити різко збільшився. Таке стрімке зростання спричинено розвитком та поширенням мережі Інтернет серед багатьох країн світу. Внаслідок цього набули популярності такі інтернет-ресурси як соціальні мережі, онлайн-магазини, онлайн-кінотеатри та різні освітні веб-сайти. Таким чином, завдяки ним маємо змогу спілкуватися з людьми зі всього світу, дізнаватися про важливу для нас інформацію, здійснювати онлайн-покупки або ж переглядати улюблені фільми.

Проте, для належного функціонування кожної з вище наведених систем світового рівня повинні використовуватися найсучасніші технології, які застосовуються при обробці величезних масивів даних. Тому, вибір правильного підходу до обробки інформації може значно покращити їх ефективність.

Наприклад, якщо розглядати онлайн-кінотеатр, то слід зазначити, що для визначення рейтингів фільмів серед глядачів можуть використовуватися певні алгоритми визначення тональності тексту.

Припустимо, що для певного фільму є дуже багато відгуків від його глядачів. Тому, самостійно визначити новому глядачу його загальний рейтинг серед тисяч відгуків буде вкрай проблематично. Через це, застосування алгоритму визначення тональності текстів для аналізу відгуків може значно пришвидшити пошуки дійсно хорошого фільму для перегляду. Це пояснюється тим, що кожному повідомленню, який оброблюється даним алгоритмом, видається оцінка його тональності у вигляді числа, яке вказує характер думки власника повідомлення про переглянутий ним фільм.

Тому, якщо ж після аналізу визначити середню оцінку серед всіх повідомлень, то можна дізнатися рейтинг даної кінокартини.

					ДП 6303.00.000 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

Таким чином, за мету виконання даної дипломної роботи була поставлена автоматизація процесів проведення аналізу тональності текстів, застосовуючи різні алгоритми та порівняння ефективності цих алгоритмів.

Реалізація даної інформаційної технології визначення тональності текстів (надалі «Інформаційна технологія») допоможе дослідникам з виявленням найкращого алгоритму, застосовуючи різні конфігурації, щоб в подальшому використовувати навчені моделі для своїх потреб.

**Практичне значення одержаних результатів.** Для реалізації інформаційної технології використано 4 підходи до аналізу тексту, а також створено зручний інтерфейс, що полегшує користувачу та адміністратору роботу з веб-застосунком даної системи.

**Публікації.** Результати роботи були опубліковані в тезах доповідей на науково-технічній конференції [1].

					ДП 6303.00.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

## 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

### 1.1 Опис предметного середовища

Як зазначено у публікації [1], на сьогоднішній день існує численна кількість різних компаній, які пропонують людям свої продукти або послуги. Для забезпечення успішного функціонування виробництва потрібно постійно проводити моніторинг ситуації на ринку, щоб задовольнити всі потреби покупців [1].

Тому, успішне просування товару на ринку потребує багатьох зусиль спеціалістів різного роду, котрі, крім проведення аналізу фінансової ситуації, повинні ще й збирати відгуки клієнтів про товар на різних сайтах соціальних мереж або інтернет-магазинах, щоб оцінити ставлення до нього [1].

Зі зростанням обсягу інформації, яку потрібно проаналізувати та обробити, аналітики зі сфери Data Science для встановлення характеру повідомлень почали використовувати аналіз тональності тексту. Ця технологія може допомогти визначити які емоції людина передала у своєму текстовому повідомленні [1].

Проте, досить часто такий аналіз зводиться лише до оцінки позитивного чи негативного відношення людей до якогось продукту або до обрахунку певних метрик [1, 7].

Таким чином, основна проблема таких методів дослідження попиту на продукцію полягає в тому, що вони не враховують контекст повідомлень користувачів, який напряду відображає ставлення покупців до даного товару або до пов'язаних з ним змін [1].

З іншого боку, з приходом нових та набагато потужніших алгоритмів здатність штучного інтелекту аналізувати текст значно покращилась. Так, використання нейронних мереж в поєднанні з глибинним навчанням і різних методів обробки та підготовки даних перетворило технологію аналізу

					ДП 6303.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

тональності повідомлень в соціальних мережах на зручний та ефективний інструмент, який значно спрощує роботу аналітика [1].

Аналіз тональності - це процес визначення того, чи є фрагмент тексту позитивним, негативним або нейтральним. Технологія аналізу тональності текстів поєднує методи обробки природної мови (ОПР) і машинного навчання для визначення зважених оцінок емоційної складової речення або фрази [1, 14].

### 1.1.1 Опис процесу діяльності

Основним процесом діяльності, який слід автоматизувати, є процес визначення тональності тексту із застосуванням різних алгоритмів.

Загалом, запуск тренування однієї моделі визначення тональності тексту та її застосування займає досить великий об'єм часу. А якщо запускати тренування кожної моделі вручну, то це займе надзвичайно великий обсяг часу. Тому, варто розглянути наступні кроки даного процесу діяльності.

**Збір даних.** Перед початком аналізу потрібно сформувати масив даних, котрий повинен бути досить великим для кращої ефективності. Зазвичай, дані збираються з популярних соціальних мереж, наприклад, з Twitter, які надають розробникам прикладний програмний інтерфейс (ППІ) для доступу до власних ресурсів та інструментарію [1, 6].

**Очищення тексту.** Засоби очищення тексту дозволяють нам підготувати дані до аналізу, видаливши стоп-слова (а, і, або, але, як, що ...), розділові знаки (коми, крапки). Ці інструменти дозволять досліднику відфільтрувати всю інформацію, яка є неважливою для аналізу [1, 6].

**Сентимент-аналіз.** На цьому етапі використовуються спеціально підібрані алгоритми для аналізу тональності зібраних даних. Такі алгоритми використовують підготовлені словники слів, кожне з яких відповідає певній емоції, щоб отримати змогу точно класифікувати повідомлення [1, 6].

					ДП 6303.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

**Оцінка результатів.** В результаті sentiment-аналізу отримується класифікація тексту, ймовірність якої була визначена обраним алгоритмом як найбільша [1].

Отже, згідно наведених вище кроків можна зрозуміти, що, наприклад, збір та очищення інформації, а також sentiment-аналіз, для якого слід провести довготривале тренування моделі на навчальній вибірці даних, дійсно займають багато часу.

Тому, при реалізації інформаційної технології визначення тональності тексту, яка включає в себе одразу декілька підходів, необхідно деякі кроки об'єднати, щоб програмний продукт міг виконувати визначення тональності тексту алгоритму набагато швидше. Через це, автоматизація запуску різних алгоритмів sentiment-аналізу відіграє ключову роль.

Таким чином, автоматизований процес діяльності можна оцінити на діаграмі діяльності з рисунку 1.1.

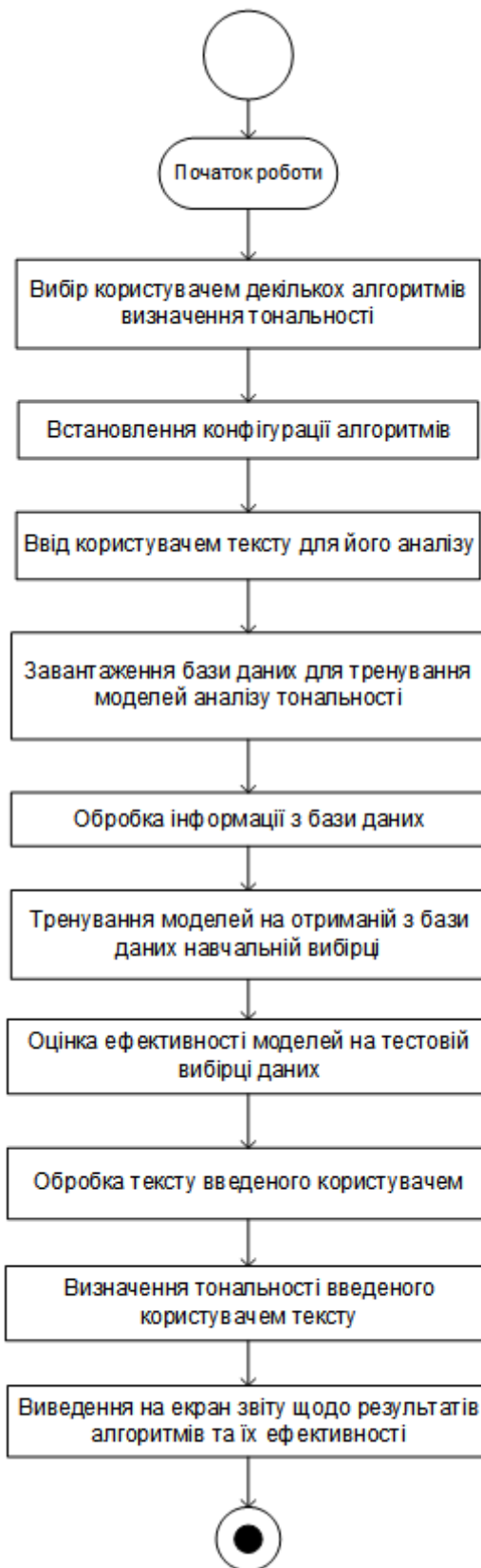


Рисунок 1.1 - Діаграма діяльності процесу визначення тональності тексту після автоматизації

Змн.	Арк.	№ докум.	Підпис	Дата

### 1.1.2 Опис функціональної моделі

Для успішного функціонування системи доступ до неї повинен бути розмежований між адміністратором та користувачем. Тому, слід виділити функції та обов'язки користувача та адміністратора в системі.

Адміністратор системи: надає користувачам доступ до моделей визначення тональності тексту або ж відкликає його.

Користувачі: отримують доступ до API розроблюваної системи та до веб-сайту системи, на якому вони можуть вибрати моделі визначення тональності тексту та здійснити запуск їх тренування.

Крім цього, користувач має змогу завантажити моделі та їх конфігурацію з бази даних.

Також, користувач може вручну задати параметри конфігурації певної моделі або ж скористатися запропонованим варіантом конфігурації.

Після того як тренування всіх моделей завершиться, користувач зможе переглянути детальний звіт щодо їх ефективності.

Користувач може задати власний текст для визначення його тональності і також побачити звіт з результатами роботи алгоритмів.

Функції Адміністратора:

- а) надання доступу користувачам до моделей;
- б) відкликання доступу у користувачів до моделей.

Функції Користувача:

- а) отримання доступу до API системи та до його веб-сайту;
- б) вибрати моделі визначення тональності тексту;
- в) запуск тренування моделей;
- г) завантаження моделей та їх конфігурації з бази даних;
- д) встановлення параметрів конфігурації;
- е) задання власного тексту для його аналізу;
- ж) перегляд звіту з результатами роботи алгоритмів та їх ефективності.

Для даного програмного продукту була побудована діаграма варіантів використання. Цю діаграму можна детальніше розглянути у графічних матеріалах до пояснювальної записки.

Відповідно до функцій програмного продукту були виявлені функціональні вимоги, список яких наведений у таблиці 1.1.

Таблиця 1.1 – Список функціональних вимог до програмного продукту

Назва функції	Функціональна вимога	Пріоритет вимоги
1	2	3
Функція завантаження бази даних текстів для тренування моделей	а) програмний продукт повинен завантажувати базу даних текстів з мережі Інтернет; 1) при відсутності доступу до мережі Інтернет програмний продукт повинен здійснити спробу підключення до завчасно збереженої бази даних	Високий Середній
Функція очистки та підготовки даних	а) з вхідних текстів повинна видалятися зайва для аналізу тональності інформація; 1) з тексту повинні видалятися розділові знаки; 2) з тексту повинні видалятися стоп-слова.	Високий Високий Середній
Функція встановлення конфігурації моделей	а) конфігурація повинна задаватися вручну; б) конфігурація повинна задаватися автоматично; 1) дані про конфігурацію повинні зберігатися у базі даних; 2) дані про конфігурацію повинні завантажуватися з бази даних.	Низький Високий Високий Високий
Функція запуску тренування моделей	а) користувач повинен отримати змогу обирати яку модель слід тренувати; б) адміністратор повинен надати користувачу доступ до інструментарію керування моделями; в) прогрес тренування моделей повинен відображатися на екрані користувача; 1) користувач повинен отримати сповіщення щодо закінчення тренування моделей; г) натреновані моделі повинні зберігатися у базі даних.	Високий Високий Середній Низький Високий

Продовження таблиці 1.1

1	2	3
Функція здійснення аналізу тональності текстів	а) Користувач повинен отримати змогу самостійно вибирати який алгоритм слід застосувати для аналізу тексту на тональність; 1) Якщо певна модель аналізу тональності не була натренована, то доступ користувачу до неї повинен бути заблокований.	Високий  Високий
Функція формування та відображення звітності	а) Звіт щодо роботи алгоритмів повинен включати певний набір показників: 1) точність класифікації; 2) швидкодія алгоритму; 3) визначений клас тексту; 4) конфігурація моделі; 5) тип алгоритму.	Високий  Високий Високий Середній Високий
Функція порівняння ефективності різних моделей аналізу тональності	а) На екран користувача повинна виводитися порівняльна таблиця ефективності алгоритмів аналізу тональності, яка складається з наборів показників зі звітів щодо кожного алгоритму б) На екран користувача повинен відображатися графік ефективності всіх алгоритмів	Високий    Середній

## 1.2 Огляд наявних аналогів

Розглянемо наступні аналоги даної інформаційної технології.

### HubSpot's Service Hub

Інструменти сервісного центру HubSpot містять інструмент зворотного зв'язку з клієнтами, який може розбити якісні відповіді опитування та оцінити їх за позитивними чи негативними намірами. Він використовує опитування NPS, щоб уточнити, чи був огляд клієнта хорошим чи поганим і організовує їх, виходячи з їх настроїв. Користувачі аналізують результати, дивлячись на одну вичерпну інформаційну панель, що включає діаграми та графіки, які забезпечують огляд задоволеності клієнтів [5].

Сервіс-центр HubSpot також може аналізувати клієнтів на індивідуальній основі. За допомогою нього можна інтегрувати свій CRM з

Service Hub та переглянути відповіді опитування від конкретних контактів у своїй базі даних [5]. Таким чином, можна швидко визначити незадоволених клієнтів та прослідкувати за ними після негативного досвіду роботи з власним брендом.

### Quick Search

Quick Search - це інструмент аналізу тональності, який є частиною більшої платформи обслуговування клієнтів під назвою Talkwalker [5]. Цей інструмент найкраще працює з каналами соціальних медіа. Тому може точно розповісти, як люди ставляться до облікових записів користувача інструменту у соціальних мережах. Швидкий пошук розглядає згадки, коментарі, залучення та інші дані, щоб надати змогу отримати купу інформації про те, як клієнти реагують на діяльність користувача у соціальних мережах [5].

### Repustate

Repustate має складний API для аналізу тексту, який досить точно оцінює тональність відгуків клієнтів. Його програмне забезпечення може підбирати текст короткої форми та сленгу [5]. Крім цього, якщо у відгуці використовуються емотикони, то Repustate визначить чи вказаний символ є позитивним або ж негативним на основі контексту цілої розмови. Repustate також дозволяє гнучко налаштувати своє API. Тому, якщо у введеному користувачем тексті зустрічається сленг або омоніми, то є можливість внести відомості про це у систему Repustate. Таким чином, користувач має повний контроль над тим, як програмне забезпечення аналізує відгуки.

### Lexalytics

Lexalytics пропонує користувачу інструмент аналізу тексту, який спеціалізується на трактуванні того, чому клієнт певним чином реагує на бізнес-користувача. Він використовує технологію обробки природньої мови для збору необхідної інформації, а потім проводить аналіз тональності для визначення відгуку клієнта [5]. Врешті-решт, Lexalytics завершує процес, перетворюючи отриману ним інформацію у зручний та легкий для читання формат.

					ДП 6303.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

### Critical Mention

Critical Mention відрізняється від інших варіантів цього списку, оскільки даний програмний продукт аналізує новини та інші публікації, в яких зустрічаються згадки про бізнес-користувача [5]. Таким чином, користувач може побачити наочну картину того, як змінюються думки людей про його продукт на основі сукупності різних новин та статей, які йдуть у хронологічному порядку.

## 1.3 Постановка задачі

### 1.3.1 Призначення розробки

Призначенням даної інформаційної технології є автоматизація процесів проведення аналізу тональності текстів, застосовуючи різні алгоритми та порівняння ефективності цих алгоритмів.

### 1.3.2 Цілі та задачі розробки

Ціллю створення даної інформаційної технології є підвищення ефективності аналізу тональності тексту.

Дану ціль можна досягти за умови, якщо програмний продукт буде виконувати наступні задачі:

- а) завантаження бази даних текстів для тренування моделей;
- б) очистка та підготовка даних;
- в) встановлення конфігурації моделей;
- г) запуск тренування моделей;
- д) здійснення аналізу тональності тексту;
- е) формування та відображення звітності;
- ж) порівняння ефективності різних моделей.

Для реалізації задач з пунктів г) та д) слід виконати наступні кроки:

- а) ознайомитися з існуючими підходами визначення тональності текстів;

					<b>ДП 6303.00.000 ПЗ</b>	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

- б) розглянути приклади застосування даних підходів;
- в) обрати декілька підходів сентиментального аналізу, які здатні дати найточнішу оцінку емоційної складової тексту;
- г) проаналізувати можливості покращення обраних підходів;
- д) розробити систему з використанням обраних алгоритмів визначення тональності тексту;
- е) створити план тренування групи моделей визначення тональності;
- ж) створити план запуску кожної моделі з групи для проведення аналізу тексту.

### Висновок до розділу

В даному розділі було описано предметне середовище інформаційної технології визначення тональності тексту та процес діяльності цієї системи. Було встановлено, що аналіз тональності тексту є досить потужним інструментом для визначення емоційної складової відгуку покупця щодо придбаного товару. Ця технологія дозволяє власникам великих компаній отримувати цінну інформацію з соціальних мереж щодо становища їх продукції на ринку.

Також, було визначено акторів у системі та описано їх функції у ній.

Крім цього, було встановлено, що автоматизація проведення визначення тональності стосуватиметься його 4 кроків: Збір даних, Очищення тексту, Сентимент-аналіз та Оцінка результатів. Внаслідок цього, за допомогою діаграми діяльності було описано процес діяльності визначення тональності тексту після автоматизації. Врешті-решт, були розглянуті існуючі аналоги даного програмного продукту та визначені їх ключові особливості.

					ДП 6303.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

## 2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1 Вхідні дані

Для даної інформаційної технології вхідні дані визначаються наступним набором параметрів:

- конфігурація моделі визначення тональності тексту;
- максимальна кількість слів у текстах навчальної та тестової вибірок;
- число найпоширеніших слів у текстах навчальної та тестової вибірок;
- навчальна вибірка даних для тренування моделей;
- тестова вибірка даних для перевірки точності класифікації;
- відмітки, що вказують на тональність текстів з навчальної вибірки;
- відмітки, що вказують на тональність текстів з тестової вибірки;
- заздалегідь збережена натренована модель визначення тональності тексту;
- вручну введений користувачем системи текст для перевірки моделей.

### 2.2 Вихідні дані

#### 2.2.1 Перелік вихідних сигналів

Для даної інформаційної технології можна навести наступний перелік вихідних сигналів:

- точність класифікації моделями текстів навчальної вибірки;
- точність класифікації моделями текстів тестової вибірки;
- значення функції втрат в результаті невірної класифікації текстів з навчальної вибірки моделями нейронних мереж;

					ДП 6303.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

- значення функції втрат в результаті невірної класифікації текстів з тестової вибірки моделями нейронних мереж;
- результат класифікації моделями вручну заданого тексту;
- файли зі збереженими в них моделями нейронних мереж;
- конфігурації моделей нейронних мереж.

### 2.2.2 Перелік вихідних документів

Всі документи, які утворюються в результаті роботи програмного продукту, поділяються на дві категорії:

- файли ієрархічного формату даних .h5 зі збереженими в них моделями нейронних мереж;
- екранні форми у вигляді звітів.

До документів з категорії б) відносяться:

- таблиця з порівняльною характеристикою ефективності кожної моделі визначення тональності тексту;
- графік зростання точності класифікації нейронними мережами і спадання величини функції втрат впродовж періоду тренування;
- гістограма точності моделей на тестовій вибірці;
- гістограма величини функції втрат при класифікації тестової вибірки нейронними мережами.

### 2.3 Опис структури бази даних

Для даної інформаційної технології була створена база даних MongoDB, в якій, крім наявних таблиць з даними про конфігурацію моделей аналізу тональності та про облікові записи користувачів, знаходиться також файлова система. В цій файловій системі зберігаються файли формату .h5, в яких містяться архітектура і ваги нейронних мереж. Структурну схему бази даних можна розглянути у графічних матеріалах до пояснювальної записки.

					ДП 6303.00.000 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

Опишемо структуру таблиць бази даних у таблиці 2.1.

Таблиця 2.1 – Опис таблиць бази даних

Назва таблиці	Назва параметру	Опис параметру	Тип даних в параметрі	Тип ключа
Configuration	_id	Унікальний номер елемента таблиці	ObjectID	PK
	ModelName	Назва конфігурації моделі	String	-
	TopWords	Число, яке встановлює мінімально можливу частість слів у текстах	Int32	-
	MaxReviewLength	Максимальна кількість слів у відгуку	Int64	-
	Configuration	Набір параметрів конфігурації, які розділені певним знаком	Array	-
	Filename	Номер файлу у таблиці fs.files	ObjectID	FK
fs.files	filename	Назва моделі	String	-
	length	Розмір моделі	Int64	-
	Content_type	Тип вмісту	String	-
	upload_date	Дата збереження	Date	-
	md5	Ідентифікаційний номер моделі	ObjectID	PK
	metadata	Метадані	String	-
Users	_id	Унікальний номер користувача	ObjectID	PK
	Login	Логін користувача	String	-
	Password	Пароль користувача	String	-
	Access	Тип доступу	Bool	-

## 2.4 Структура масивів інформації

Конфігурація моделі аналізу тональності тексту являє собою список параметрів різного роду, кількість яких варіюється в залежності від типу моделі.

Оскільки модель наївного баєсового класифікатора не потрібно конфігурувати, то загальна кількість конфігурацій для решти моделей

дорівнює трьом. Таким чином, у таблицях 2.2, 2.3 та 2.4 приведемо опис параметрів, які входять у конфігурації моделей.

Таблиця 2.2 – Конфігурація до нейронної мережі ДКЧП

Найменування параметру	Позначення	Формат
Довжина векторного представлення тексту	embed_vec_len	Ціле невід’ємне число
Маска для нульових елементів векторного представлення тексту	mask_zero	Булеве значення
Кількість прихованих нейронів у шарі ДКЧП	units	Ціле невід’ємне число
Частка нейронів, які не будуть включені в процес навчання	dropout	Дробове невід’ємне число
Ймовірність ігнорування нейроном вхідних сигналів з попереднього стану	recurrent_dropout	Дробове невід’ємне число
Кількість нейронів у звичайному шарі нейронної мережі	dense_units	Ціле невід’ємне число
Розмір вибірки, для здійснення однієї ітерації тренування моделі	bat_size	Ціле невід’ємне число
Кількість періодів навчання нейронної мережі	max_epochs	Ціле невід’ємне число

Таблиця 2.3 – Конфігурація до згорткової нейронної мережі

Найменування параметру	Позначення	Формат
1	2	3
Довжина вихідних векторів з шару Embedding	embedding_size	Ціле невід’ємне число
Розмір вікна згортки	kernel_size	Ціле невід’ємне число
Кількість фільтрів для аналізу векторів слів	filters	Ціле невід’ємне число
Розмір вікна субдискретизації	pool_size	Ціле невід’ємне число
Кількість нейронів у першому звичайному шарі нейронної мережі	dense_units1	Ціле невід’ємне число

Продовження таблиці 2.3

1	2	3
Кількість нейронів у другому звичайному шарі нейронної мережі	dense_units2	Ціле невід’ємне число
Розмір вибірки, для здійснення однієї ітерації тренування моделі	batch_size	Ціле невід’ємне число
Кількість періодів навчання нейронної мережі	epochs	Ціле невід’ємне число

Таблиця 2.4 – Конфігурація до комбінованої нейронної мережі

Найменування параметру	Позначення	Формат
Довжина вихідних векторів з шару Embedding	embedding_size	Ціле невід’ємне число
Розмір вікна згортки	kernel_size	Ціле невід’ємне число
Кількість фільтрів для аналізу векторів слів	filters	Ціле невід’ємне число
Розмір вікна субдискретизації	pool_size	Ціле невід’ємне число
Кількість LSTM елементів	lstm_output_size	Ціле невід’ємне число
Розмір вибірки для здійснення однієї ітерації тренування моделі	batch_size	Ціле невід’ємне число
Кількість періодів навчання нейронної мережі	epochs	Ціле невід’ємне число
Частка нейронів, які не будуть включені в процес навчання	dropout	Дробове невід’ємне число
Розмір кроку для зсуву вікна згортки	strides	Ціле невід’ємне число
Кількість нейронів у звичайному шарі нейронної мережі	dense	Ціле невід’ємне число

Опишемо детальніше деякі вказані вище параметри.

Довжина векторного представлення тексту впливає на об’єм інформації, яка буде стосуватися певного слова у тексті. Тому, чим більше це число, тим більший буде вектор представлення слова і тим більше нейронів будуть залучені в процес аналізу ознак цього слова.

При встановленні маски на нульові елементи векторного

представлення тексту ми присвоюємо певне значення для нульових елементів вектору, щоб при проведенні обчислень всередині нейронної мережі ці значення ігнорувалися. Таким чином, це може призвести до пришвидшення обчислень або ж до отримання більшої точності класифікації через відсутність «інформаційного шуму».

При визначенні частки нейронів, які не будуть використовуватися під час навчання, ми зможемо зменшити ефект від перенавчання нейронної мережі в тому випадку, коли її точність вже досягла свого найбільшого значення. При появі такого ефекту нейронна мережа перестає виявляти нові, більш приховані ознаки, оскільки її нейрони, при обробці інформації, спираються на «оптимальні» вагові коефіцієнти попередніх нейронів.

Тому, виключення з нейронної мережі деяких елементів дасть можливість активним нейронам самостійно вдосконалювати свої коефіцієнти, що призведе до поступового підвищення точності моделі.

Для вказаних вище конфігурацій у програмному продукті створюються спеціальні класи, в якому у вигляді полів зберігається вміст цих наборів даних. Конструктори даних класів для їх ініціалізації приймають на вхід список параметрів, який має наступний вигляд:

*(Параметр 1, Параметр 2, ..., Параметр N).*

Після цього, кожному полю класу конфігурації присвоюється значення відповідного параметра.

Навчальна і тренувальна вибірки разом із відмітками про клас текстів завантажуються із файлу `imdb.npz`, який містить дані про 25000 позитивних і негативних відгуків [8] користувачів сайту <https://www.imdb.com/> щодо переглянутих фільмів.

Цей файл являє собою архів, в якому містяться файли формату `.npz`:

- а) `x_test.npz` – містить тестовий набір даних з 25000 позитивними і негативними відгуками;

										Арк.
										23
Змн.	Арк.	№ докум.	Підпис	Дата	ДП 6303.00.000 ПЗ					

- б) *x\_train.npy* – містить тренувальний набір даних з 25000 позитивними і негативними відгуками;
- в) *y\_test.npy* – містить 25000 відміток, щодо класів відгуків з тестової вибірки;
- г) *y\_train.npy* – містить 25000 відміток, щодо класів відгуків з навчальної вибірки.

Тестовий та тренувальний набори даних, які збережені у файлах з пунктів а) і б) мають наступний вигляд:

*[[Частота слова 1, Частота слова 2, ..., Частота слова N1],*  
*[Частота слова 1, Частота слова 2, ..., Частота слова N2],*

...,

*[Частота слова 1, Частота слова 2, ..., Частота слова N25000]].*

Також, слід зауважити, що під частотою слів у даному прикладі мається на увазі загальна частота певного слова у всьому корпусі текстів.

До кожного відгуку стосовно фільму відповідає певна відмітка, яка вказує чи позитивний цей відгук чи ні. Формат даних, за допомогою якого ці відмітки зберігаються у файли з пунктів в) і г) можна представити наступним чином:

*[Тональність 1, Тональність 2, ..., Тональність 25000].*

Тональність тексту у даному наборі даних може позначатися 1, якщо відгук позитивний, або 0 – для негативного відгуку.

Загалом, структура збереженої нейронної мережі у файлі формату .h5 складається з двох частин:

- а) архітектура нейронної мережі;
- б) ваги нейронної мережі.

Оскільки дані, які збережені у даному файлі, відображаються у бінарному вигляді, то для повного огляду його структури необхідно скористатися спеціальною програмою HDF5 [10].

Проте, архітектура нейронної мережі може бути також представлена у вигляді JSON або в YAML [4]. Тому, наведемо нижче загальне представлення архітектури нейронної мережі у форматі YAML:

*Платформа запуску нейронної мережі: tensorflow*

*Клас моделі: Клас 1.*

*Конфігурація:*

*Шари:*

*- Назва шару: Шар 1.*

*Параметри шару:*

*Параметр 1: значення параметру 1;*

*Параметр 2: значення параметру 2;*

...

*Параметр N: значення параметру N.*

*- Назва шару: Шар 2.*

*Параметри шару:*

*Параметр 1: значення параметру 1;*

*Параметр 2: значення параметру 2;*

...

*Параметр N: значення параметру N;*

...

*- Назва шару: Шар M.*

*Параметри шару:*

*Параметр 1: значення параметру 1;*

*Параметр 2: значення параметру 2;*

...

*Параметр N: значення параметру N.*

*Назва моделі: Модель 1.*

*Номер версії бібліотеки Keras: Приклад(2.3.1).*

					ДП 6303.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

### Висновок до розділу

В даному розділі я зміг визначити перелік вхідних та вихідних даних. Так, основними параметрами, які подаються на вхід є конфігурація моделей визначення тональності тексту, навчальна та тренувальна вибірки даних із проставленими відмітками, що вказують на клас тексту. До речі, не менш важливими є такі параметри, як максимальна кількість слів та число найпоширеніших слів у текстах навчальної та тестової вибірок, оскільки вони дозволяють отримати вибірки, які пришвидшать навчання

Крім цього, в якості вихідних даних інформаційної технології є результати щодо класифікації вхідних текстів та точність визначення цих класів. Також, ці результати можуть бути представлені у вигляді у таблиць, гістограм та графіків. Врешті-решт, після тренування нейронних мереж їх моделі зберігаються у спеціальних файлах ієрархічного формату даних .h5, які, в свою чергу, зберігаються у файловій системі бази даних MongoDB.

Конфігурація моделей визначення тональності також зберігається у базу даних. Дані про неї збережені у вигляді таблиці, в якій містяться всі необхідні параметри для майбутнього запуску моделей.

### 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

#### 3.1 Змістовна постановка задачі

Користувачами даного програмного продукту можуть бути люди, які є власниками великих компаній, що випускають масу продукції на продаж, або ж звичайні аналітики, які бажають отримати зручний інструментарій для аналізу текстів на тональність. Так, власників зазвичай цікавить, що думають покупці про їх товар, оскільки від цього напряму залежить обсяг їх прибутку. Тому, якщо керівництво компанії усвідомить, що, наприклад, клієнти незадоволені наданими їм послугами або отриманим продуктом, то його рівень продажів може різко знизитися.

Через це, даний програмний продукт з реалізованою інформаційною технологією визначення тональності текстів повинен надавати можливість її користувачу вводити текст відгуку, щоб потім, після завершення роботи програми, він зміг отримати детальний звіт щодо тональності тексту.

Крім цього, аналітики, які хочуть дослідити зміну тональності тексту або ж ефективність реалізованих алгоритмів в програмі, повинні отримати змогу вводити різні параметри моделі для її конфігурації. Також, вони зможуть обирати зі списку наявні алгоритми для порівняння їх швидкодії та точності на графіках, таблицях та гістограмах.

Варто зазначити, що для визначення тональності тексту необхідно мати навчальну та тестову вибірку текстів. Навчальна вибірка буде використовуватися для «навчання» моделей. Тобто алгоритм, згідно якого працює модель, повинен здійснювати обробку навчальних даних для встановлення певних взаємозв'язків, закономірностей та ключових ознак або відмінностей між словами або ж цілими реченнями. Після того, як тільки тренування завершилося, навчену модель потрібно перевірити на вибірці з тестових даних, щоб встановити точність класифікації текстів та загальну її ефективність в тій чи іншій ситуації. Також, варто оцінити час проведення

					ДП 6303.00.000 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

процесу підготовки моделі до її подальшого застосування. Так користувач дізнається який з алгоритмів працює найшвидше, а який є найповільнішим.

Виходячи зі всього вище наведеного, можна сформулювати основну задачу даної інформаційної технології як задачу визначення тональності тексту.

### 3.2 Математична постановка задачі

Сформулюємо математичну постановку задачі наступним чином:

нехай

- $X_{навч}$  – це множина текстів навчальної вибірки;
- $X_{тест}$  – це множина текстів з тестової вибірки;
- $Y_{навч}$  – множина відміток тональності текстів з  $X_{навч}$ ;
- $Y_{тест}$  – множина відміток тональності текстів з  $X_{тест}$ ;
- $x_{кор}$  – це текст введений користувачем;
- $y_{кор}$  – це реальне значення тональності тексту  $x_{кор}$ ;
- $S$  – словник частот слів з текстів  $X_{навч}$  та  $X_{тест}$ ;
- $n_{навч}$  – розмір вибірки  $X_{навч}$ ;
- $n_{тест}$  – розмір вибірки  $X_{тест}$ ;
- $e$  – це кількість періодів навчання моделі;
- $f_{min}$  – це мінімальна частота слова  $w$  у словнику  $S$ .

То для вирішення задачі визначення тональності тексту необхідно знайти значення тональності тексту, класифікованого моделлю  $m$ .

Це значення ми позначимо грецькою літерою  $\gamma$ .

Цільовою функцією у поставленій задачі – є мінімізація різниці між очікуваною тональністю та одержаною за допомогою моделі  $m$ .

Тому, представимо цю функцію у наступному вигляді:

$$ЦФ: d = \min |y_{кор.} - \gamma|. \quad (3.1)$$

					ДП 6303.00.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.3 Обґрунтування методу розв'язання

Для розв'язання задачі визначення тональності текстів були обрані такі методи:

- наївний баєсів класифікатор;
- рекурентна нейронна мережа з довгою короткочасною пам'яттю (ДКЧП);
- згорткова нейронна мережа (ЗНМ);
- комбінація ЗНМ та ДКЧП.

Використання наївного баєсового класифікатора обумовлене тим, що його «наївність» дозволяє не звертати увагу на можливі зв'язки між словами, тобто ми припускаємо, що всі слова, які зустрічаються у тому чи іншому тексті є незалежними між собою. Тому, використання даного методу не тільки значно полегшує процес вирішення поставленої задачі, а й прискорює одержання результату.

Оскільки на сьогоднішній день спектр використання нейронних мереж є дуже великим, то для аналізу текстів вони застосовуються також.

Тому, спеціально для задачі визначення тональності тексту була обрана рекурентна нейронна мережа з технологією довгої короткочасної пам'яті (ДКЧП). Рекурентність цієї мережі полягає у використанні в обчисленнях значень з попередніх обрахунків. Але, через це виникає її головний недолік, який вирішується за допомогою ДКЧП, щоб «забувати» з часом застарілу інформацію. Тому, реалізація даної технології у задачі визначення тональності текстів дозволить зберігати у «пам'яті» нейронної мережі інформацію про закономірності та взаємозв'язки між словами, що підвищить ефективність рекурентної нейронної мережі.

Загалом, згорткові нейронні мережі зазвичай використовуються для розпізнавання зображень. Їх ключова особливість полягає в тому, що вони

розділяють вхідне зображення на безліч різних шаблонів з груп пікселів для подальшої класифікації цього зображення. Тому, використання згорткових нейронних мереж для класифікації тексту та визначення його тональності може бути досить ефективним рішенням.

Врешті-решт, поєднання переваг рекурентної та згорткової нейронних мереж для вирішення поставленої задачі може показати набагато кращий результат ніж при використанні кожної з них окремо. Тому даний підхід теж вартий його реалізації у інформаційній технології.

### 3.4 Опис методів розв'язання

#### 3.4.1 Наївний баєсів класифікатор

Основною ідеєю наївного баєсівського класифікатора є обчислення ймовірностей присвоєння тексту якогось класу, використовуючи спільні ймовірності слів і класів [1].

Сформулюємо теорему Баєса за допомогою співвідношення (3.2) [1]:

$$P(C_k | x_1, \dots, x_n) = \frac{P(C_k)P(x_1, \dots, x_n | C_k)}{P(x_1, \dots, x_n)}, \quad (3.2)$$

де  $C_k$  - клас тексту, а  $(x_1, \dots, x_n)$  - це його залежний вектор ознак [1, 14].

Згідно до «наївних» припущень щодо умовної незалежності, для даного класу  $C_k$  кожна ознака вектору  $x_i$  умовно незалежна від будь-якої іншої ознаки  $x_j$  при  $i \neq j$  [1, 14].

$$P(x_i | C_k, x_1, \dots, x_n) = P(x_i | C_k). \quad (3.3)$$

Таким чином, відношення (3.3) може бути спрощено до [1]:

$$P(x_i | C_k, x_1, \dots, x_n) = \frac{P(C_k) \prod_{i=1}^n P(x_i | C_k)}{P(x_1, \dots, x_n)}. \quad (3.4)$$

Оскільки  $P(x_1, \dots, x_n)$  є константою, то якщо значення змінних ознак відомі, можна використовувати правило класифікації (3.5) з логарифмічними значеннями ймовірностей для уникнення арифметичного переповнення [1]:

$$\hat{y} = \arg \max_k (\ln P(C_k) + \sum_{i=1}^n \ln P(x_i | C_k)). \quad (3.5)$$

Поліноміальний розподіл параметризується вектором  $\theta_k = (\theta_1, \dots, \theta_n)$  для кожного класу  $C_k$ , де  $n$  - це кількість ознак, а  $\theta_{ki}$  - ймовірність  $P(x_i | C_k)$  ознаки  $i$  з'являється у зразку, котрий належить до класу  $C_k$  [1, 14].

Параметри  $\theta_k$  оцінюються відносним підрахунком частоти за допомогою співвідношення (3.6) [1]:

$$\hat{\theta}_{ki} = \frac{N_{ki} + \alpha}{N_k + \alpha n}, \quad (3.6)$$

де  $N_{ki}$  - це число разів, коли об'єкт  $i$  з'являється у прикладі класу  $k$  в навчальному наборі  $T$ , а  $N_k$  - це загальна кількість всіх об'єктів класу  $C_k$  [1, 14].

Для запобігання появи нульових ймовірностей, коли якийсь слово відсутнє у навчальній вибірці, застосовується згладжування Лапласа, яке у формулі (3.6) позначається змінною  $\alpha$ , яка зазвичай дорівнює 1 [1].

Зрештою, формула (3.5) набуває наступного вигляду [1]:

$$\hat{y} = \arg \max_k (\ln P(C_k) + \sum_{i=1}^n \ln \frac{N_{ki} + \alpha}{N_k + \alpha n}). \quad (3.7)$$

Таким чином, згідно формули (3.7) для всіх класів обчислюються їх ймовірності. Після цього, текст класифікується тим класом, який матиме найбільшу ймовірність [1].

### 3.4.2 Рекурентна нейронна мережа з ДКЧП

Рекурентні нейронні мережі - це тип нейронних мереж, де входи прихованих шарів в поточний момент часу залежать від попередніх виходів прихованого шару [1, 3].

Оскільки часова послідовність в РНН зростає, то значення вагових коефіцієнтів можуть неконтрольно збільшуватися або зменшуватися до нуля. Щоб впоратися з проблемою зникаючого градієнта при навчанні звичайних РНН, було запропонована технологія довгої короткочасної пам'яті (ДКЧП), яка призначалася для вивчення довгострокової залежності на тривалому часовому проміжку. Таким чином, за допомогою ДКЧП до вхідних і вихідних блоків або «вентилів» додається ще й «забувальний вентиль». Зазвичай, нейронні мережі з такою архітектурою використовуються для прогнозування часових рядів і розпізнавання рукописного тексту [1, 3].

Для обробки природної мови є корисним проаналізувати кількість входження кожного слова у документі. Найпростіше це зробити за допомогою унітарного кодування, яке дозволяє представити входження кожного слова у документі у вигляді бінарного вектору. Серед різних моделей векторних представлень слів найбільш популярними з них є Word2Vec і GloVe [1, 3].

Після представлення кожного слова відповідним йому вектором, навченим згідно моделі Word2Vec, слова  $\{T_1, \dots, T_n\}$  вводяться у ДКЧП один за одним, як показано на рисунку 3.1 [1].

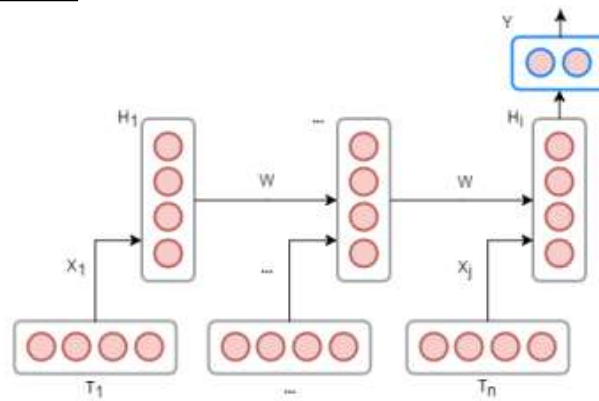


Рисунок 3.1 - Основна ідея ДКЧП [5, 6]

На рисунку 1 кожна частинка  $T_i$  спершу перетворюється у відповідний вхідний вектор  $x_i$ , використовуючи модель Word2Vec, а потім входить у ДКЧП один за одним. Далі, в кожен момент часу  $j$  вихід  $W$  прихованого шару  $H_i$  буде переданий назад у прихований шар разом із наступним входом  $x_{j+1}$  у наступному моменті часу  $j+1$ . Зрештою, останній вихід  $W_n$  «згодовується» вихідному шару [1, 3].

Щоб діяти за правилами послідовного введення ДКЧП ми спочатку конвертуємо текст у тривимірну матрицю  $M(X, Y, Z)$ , де  $X$  - розмірність моделі вкладення слів Word2Vec,  $Y$  - кількість слів у тексті, а  $Z$  - кількість текстів. Кількість нейронів у вхідному шарі відповідає розміру моделі Word2Vec, а число нейронів у вихідному шарі дорівнює кількості класів. За допомогою зворотного поширення у часі відбувається коригування вагового коефіцієнту ребер прихованого шару в кожен момент часу. Після завершення кількох періодів навчання нейронної мережі ми отримуємо готову модель для класифікації тексту [1, 3].

Після навчання моделі кожен текст з тестової вибірки проходить таку ж саму процедуру, як і елементи навчальної вибірки. Далі, всі процеси, які відображені на рисунку 1 повторюються, але цього разу ваги ребер не оновлюються [1, 3]. Потім, отримані результати класифікації текстів можна порівняти із реальними, щоб визначити точність готової моделі [1].

### 3.4.3 Згорткова нейронна мережа

Згорткові нейронні мережі відрізняються в першу чергу від звичайних тим, що не всі їх нейрони з'єднані між собою. Замість цього, над вхідним шаром нейронної мережі виконується згортка, щоб обчислити вихідне значення. В контексті аналізу текстів згорткою є ковзаюче вікно, яке рухається вздовж векторного представлення слова.

Головною особливістю даної нейронної мережі є накладення різноманітних фільтрів в кожному її шарі для даних, які проходять крізь ці шари. Так, цих фільтрів повинно бути дуже багато, щоб ретельно обробити всю інформацію. Кожен фільтр у шарі відповідає за обробку своєї індивідуальної ознаки. Після застосування у одному шарі багатьох фільтрів для розрахунку його вихідного значення результати цих фільтрів поєднуються

Першим шаром даної мережі є шар векторного представлення слова. В цей шар входять тексти, які перед цим пройшли спеціальну обробку.

Суть даної обробки полягає в тому, що всі слова у тексті замінюються на числові значення, що позначають частоту цих слів у словнику корпусу текстів з навчальної вибірки.

Для того, щоб нейронна мережа могла працювати не з текстом, а з його словами, для кожного слова утворюється векторне представлення слова, що символізує вираження його семантики у вигляді числа. Таким чином, слова з приблизно однаковим значенням матимуть схожі вектори.

Основними параметрами цього шару є розмір батчу, тобто кількість текстів, яка буде оброблятися в шарі одночасно. Також, для того, щоб довжина вхідних текстів була однаковою задається також параметр максимальної довжини.

Далі, після шару векторного представлення слова йде згортковий шар,

					ДП 6303.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

параметрами якого є розмір фільтрів, їх кількість та крок, який вони роблять при обробці векторів слів. При обробці одного з речень тексту з векторів слів формується матриця, по якій і ковзає вікно згортки або ж фільтру.

Таким чином, на вміст даної матриці накладаються вказані фільтри, а потім обчислені результати потрапляють у агрегувальний шар, де для вихідного значення з кожного фільтру речення виконується операція максимізації, а потім вихідні вектори об'єднуються у наступному шарі, утворюючи лише один вектор для кожного речення.

Потім, ці дані потрапляють у вихідний шар, який являє собою шар звичайної нейронної мережі, де для них обчислюється наступний вираз:

$$xW + b, \quad (3.8)$$

де  $x$  - вхідний вектор речення;

$W$  - матриця вагових коефіцієнтів;

$b$  - значення нейрону зміщення.

Результатом цього виразу є ймовірності двох класів (позитивний або негативний).

#### 3.4.4 Комбінація згорткової та рекурентної нейронної мережі

В даній нейронній мережі до шарів рекурентної нейронної мережі додається також два шари зі згорткової нейронної мережі, які розміщуються після шару векторних представлень слів:

- одновимірний згортковий шар;
- агрегувальний максимізуючий шар.

**Висновок до розділу**

В даному розділі були наведені змістовна та математичні постановки задачі визначення тональності тексту. Також, було зазначено, які функції повинен виконувати програмний продукт для вирішення поставленої задачі. Далі, було встановлено основні методи виконання даної задачі. Такими методами є :

- наївний басів класифікатор;
- рекурентна нейронна мережа з ДКЧП;
- згорткова нейронна мережа;
- комбінація ДКЧП і ЗНМ.

Дані методи є досить ефективними, оскільки поєднання їх переваг у інформаційній технології дає можливість отримати потужний інструментарій, яким зможуть користуватися аналітики для проведення різноманітних досліджень, а також підприємці або ж власники великих компаній для аналізу відгуків від клієнтів про їх продукцію.

## 4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 4.1 Засоби розробки

Для реалізації даної інформаційної технології були застосовані наступні засоби розробки:

- Python;
- Keras;
- Tensorflow;
- Mathplotlib;
- Flask;
- NumPy;
- GridFS;
- MongoDB;
- Flask\_Table.

Python – це об’єктно орієнтована мова програмування високого рівня з інтегрованою динамічною семантикою для розробки веб-сайтів та застосунків [16].

Keras – це API для глибокого навчання нейронних мереж, який написаний на мові програмування Python та працює на базі платформи машинного навчання Tensorflow [2].

Mathplotlib – це всеохоплююча бібліотека, яка призначена для створення різноманітних графіків у Python [9].

Flask – це веб-фреймворк, який пропонує розробнику інструментарій, різні бібліотеки та технології, які дозволять створити власний веб-застосунок [17].

NumPy – це пакет загального призначення для обробки масивів, який надає розробнику об’єкт багатовимірного масиву з швидким доступом до нього. Крім цього, також надається спеціальний інструментарій для роботи з

					ДП 6303.00.000 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

цим об'єктом.[12]

GridFS – це специфікація MongoDB, або ж файлова система, яка призначена для збереження та отримання файлів великого розміру [11].

MongoDB – це база даних, яка використовує документ-орієнтовану модель бази даних, що підтримує різні форми даних [13].

Flask\_Table – це бібліотека, яка надає зручний інструментарій для структурування даних у вигляді таблиці та їх відображення на сторінці веб-сайту.

## 4.2 Вимоги до технічного забезпечення

### 4.2.1 Загальні вимоги

Склад технічних засобів визначається наявністю ЕОМ, яка має задовольняти наступні вимоги:

- наявність процесора з тактовою частотою не менше 1ГГц;
- наявність оперативної пам'яті місткістю не менше 2 ГБ;
- наявність жорсткого диску з вільним об'ємом пам'яті не менше 1 ГБ;
- встановлена операційна система версії Windows 7 або вище;
- наявність підключення до мережі Інтернет;
- наявність монітору.

## 4.3 Архітектура програмного забезпечення

### 4.3.1 Діаграма класів

Для відображення структури розробленого програмного продукту була створена діаграма класів. Детальніше її можна розглянути в графічному матеріалі до пояснювальної записки.

										Арк.
										38
Змн.	Арк.	№ докум.	Підпис	Дата	ДП 6303.00.000 ПЗ					



відбувається безпосередній запуск моделі, тобто її підготовка до аналізу тексту, введеного користувачем. Потім навчена модель та її конфігурація зберігаються в базу даних, а результат класифікації передається назад в клас WebSite. Врешті-решт, всі результати виводяться на веб-сторінку у вигляді звіту, який вже може переглянути користувач.

### 4.3.3 Діаграма компонентів

Для того, щоб краще відобразити яким чином взаємодіють між собою окремі структурні елементи програмного продукту була створена діаграма компонентів, яку можна розглянути на рисунку 4.1.

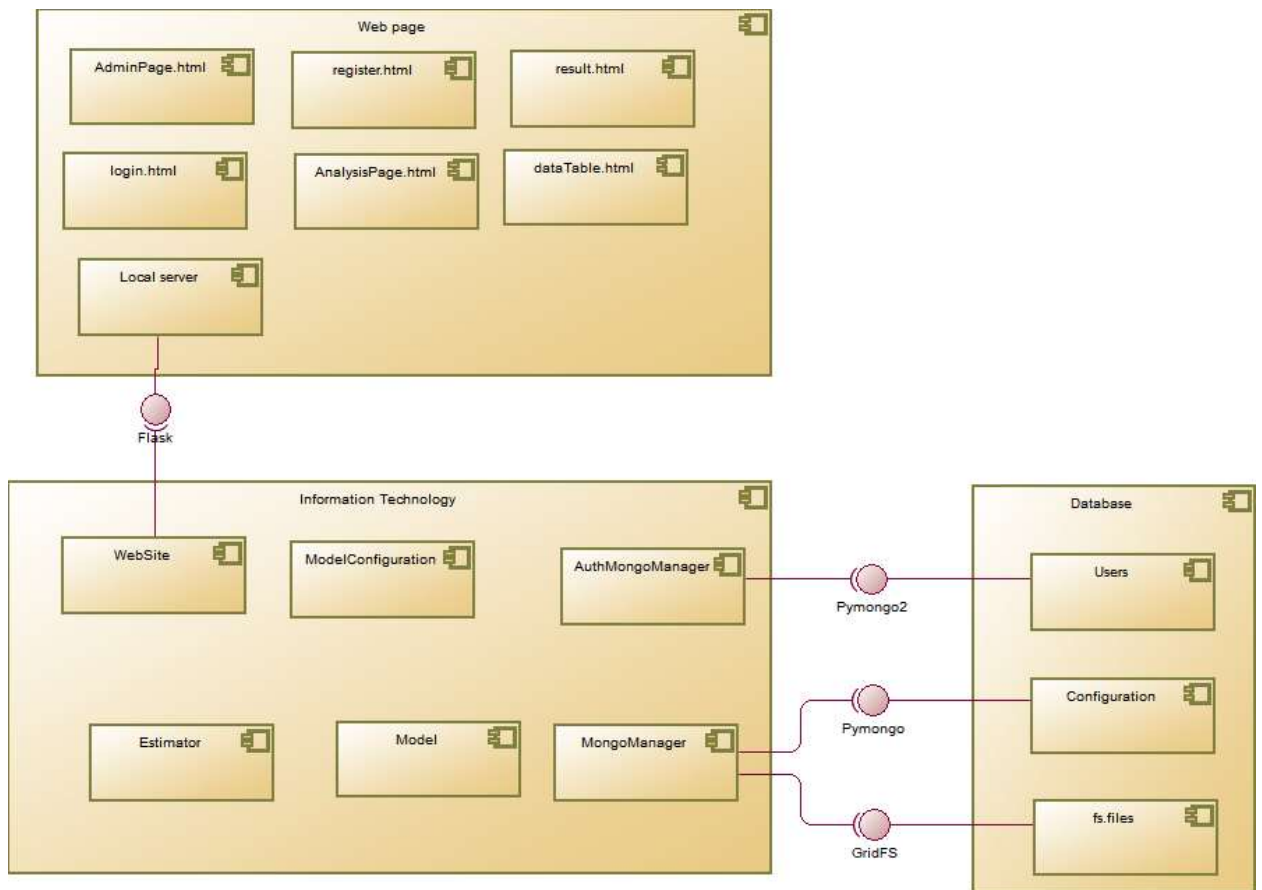


Рисунок 4.1 - Діаграма компонентів розробленого програмного продукту

Згідно наведеної вище діаграми можна побачити, що інтерфейсом взаємодії компоненту інформаційної технології та веб-сайту, на якому відображаються всі результати програми, виступає веб-фреймворк Flask.

Крім цього, за обмін даними з базою даних MongoDB відповідають gridfs та pymongo.

#### 4.3.4 Специфікація функцій

Наведемо у таблиці 4.1 специфікацію основних функцій з класів програмного продукту.

Таблиця 4.1 - Специфікація функції

Назва класу	Назва функції	Призначення функції
1	2	3
WebSite	index()	Запускає веб-сайт
	result()	Обробляє введену інформацію з веб-сайту, викликає клас Estimator для запуску моделей та повертає результат класифікації на веб-сторінку
	login():	Здійснює процедуру авторизації користувача
	register()	Здійснює процедуру реєстрації користувача
	change(id)	Змінює доступ користувача до моделей
	getConfiguration(userInputConfig, params_number)	Перетворює конфігурацію моделі, введену користувачем, у масив параметрів конфігурації
	loadData()	Завантажує моделі та їх конфігурацію з бази даних
	choose(id)	Отримує модель та її конфігурацію вибрані користувачем
	runModels():	Здійснює запуск моделей для аналізу тексту
	allconfigsConvertToType(lstmConfig, cnn_config, combined_config)	Типізує рядкові масиви параметрів конфігурацій

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 4.1

1	2	3
Estimator	estimate(self, top_words, max_words_number, LSTM_Config, CNN_Config, CNN_LSTM_Config, userText)	Запускає оцінювання моделей визначення тональності
	showTestResultsGraph(self, accList, lossList, plot)	Функція створення графіку ефективності алгоритмів на тестовій вибірці
	outputGraph(self, histories)	Функція створення графіку ефективності моделей впродовж тренування
	runLSTM(self, top_words, max_words_number, LSTM_Config, userText)	Функція запуску рекурентної нейронної мережі
	runCNN(self, top_words, max_words_number, CNN_Config, userText)	Функція запуску згорткової нейронної мережі
	runCNN_LSTM(self, top_words, max_words_number, CNN_LSTM_Config, userText)	Функція запуску комбінованої нейронної мережі
	runBayes(self, top_words, max_words_number, userText)	Функція запуску наївного баєсівого класифікатора
	makeReport(self, histories, accList, lossList, timeResults)	Функція формування звіту
	estimateLSTM(self, top_words, review_len, LSTM_Config, userText, test_x, test_y, train_x, train_y)	Функція запуску навчання рекурентної нейронної мережі та оцінки її ефективності
	estimateCNN(self, top_words, review_len, CNN_Config, userText, test_x, test_y, train_x, train_y)	Функція запуску навчання згорткової нейронної мережі та оцінки її ефективності
	estimateCNN_LSTM(self, top_words, review_len, CNN_LSTM_Config, userText, test_x, test_y, train_x, train_y)	Функція запуску навчання комбінованої нейронної мережі та оцінки її ефективності
	estimateBayes(self, top_words, review_len, userText, training_set, training_labels, validation_set, validation_labels)	Функція запуску навчання наївного баєсівого класифікатора та оцінки його ефективності
	runAll(self, filenames, review_len_s, userText)	Функція запуску аналізу тексту всіма моделями

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 4.1

1	2	3
Model	init__(self,*params)	Ініціалізує модель
	defineModel(self, test_x, test_y, train_x, train_y)	Визначає структуру моделі
	runModel(self, model,userText)	Запускає модель для здійснення прогнозу
	loadModel(self)	Завантажує модель з бази даних
	saveModel(self, model)	Зберігає модель у базу даних
LSTM, CNN, CNN_LSTM	evaluateModel(self, model, test_x, test_y)	Оцінює ефективність моделі на тестовій вибірці
	doPrediction(self, model, userText, max_review_len)	Здійснює класифікацію тексту навченою моделлю
NaiveBayes Classifier (Model)	evaluate_predictions(self,validation_set, validation_labels, verbose)	Функція формування оцінок класифікації моделей
	train(self, training_set, training_labels, alpha=1)	Функція тренування наївного Баєсівого класифікатора
	predict(self, test_doc)	Функція класифікації тексту
Configuration	__init__(self, *params)	Ініціалізація конфігурації моделі
	getConfig(self)	Функція отримання параметрів конфігурації
	getConfigAsString(self,*params)	Функція формування рядку параметрів
Mongo Manager	saveToDB(filename, model)	Функція збереження моделі у базу даних
	loadModelFromDB(filepath)	Функція завантаження моделі з бази даних
	saveConfiguration(modelName, top_words, review_len, model_config, filename)	Функція збереження конфігурації у базу даних
	loadConfiguration(modelName)	Функція завантаження конфігурації з бази даних
	makeModelTable(data)	Функція формування таблиці з моделями
	makeResultTable(data)	Функція формування таблиці з результатом класифікації
	makeReportTable(data):	Функція формування звітної таблиці

Продовження таблиці 4.1

1	2	3
UsersTable	makeUserTable(userList)	Функція формування таблиці користувачів системи
AuthMongoManager	saveUser(login,password)	Функція збереження створеного облікового запису користувача у базу даних
	loadUser(inputLogin)	Функція завантаження даних про користувача з бази даних
	loadAllUsers()	Функція завантаження списку користувачів системи з бази даних
	updateUserAccess(id,access)	Функція оновлення доступу користувача до моделі
	findRecordById(id)	Функція пошуку користувача у базі даних за ідентифікаційним номером
DataLoader	loadData(top_words, max_review_len)	Функція завантаження навчальної та тестової вибірок відгуків глядачів
	getIMDB_dataForBayes(top_words, maxlen,train_data_raw, train_labels, test_data_raw, test_labels)	Функція перетворення тестової та навчальної вибірок відгуків глядачів у придатний для баєсівого класифікатора вигляд
	loadBayesData(top_words, max_words_number,test_x, test_y, train_x, train_y)	Функція отримання даних для найвного баєсівого класифікатора

#### 4.4 Опис звітів

В результаті роботи програмного продукту на виході ми отримуємо детальний звіт щодо ефективності алгоритмів. Цей звіт представлений результатами визначення тональності введеного користувачем тексту, а також таблицею зі статистикою навчання та тестування моделей, графіком і гістограмою ефективності моделей аналізу тональності впродовж їх навчання.

На рисунку 4.2 представлений результат класифікації тексту всіма алгоритмами.

Результати роботи моделей визначення тональності тексту:

Введений користувачем текст: The movie was awesome. I love it

Назва моделі	Значення тональності	Визначений клас тексту
Рекурентна нейронна мережа з довгою короткочасною пам'яттю (ДКЧП)	0.689	Позитивний
Згортоква нейронна мережа	0.487	Негативний
Комбінована нейронна мережа(ДКЧП+ЗНМ)	0.822	Позитивний

Рисунок 4.2 - Результат класифікації 4 алгоритмами.

Також, на рисунку 4.3 можна побачити фрагмент таблиці із зібраною статистикою вказаних моделей впродовж періоду навчання та тестування.

Згортоква нейронна мережа		Назва етапу	Статистика моделі			21.682902812957764
		Навчання	Номер епохи	Точність моделі	Значення функції втрат	
			0	0.73016	0.4842801654112998	
			1	0.81106	0.26686657924003604	
			2	0.92556	0.19378862264900207	
		Тестування	Номер епохи	Точність моделі	Значення функції втрат	
			0	0.85052	0.3596198750508133	

Рекурентна нейронна мережа з ДКЧП		Назва етапу	Статистика моделі			198.97327164677424
		Навчання	Номер епохи	Точність моделі	Значення функції втрат	
			0	0.7638	0.5079068775939942	
			1	0.85502	0.34953057063129886	
			2	0.87048	0.31905301440980668	
		Тестування	Номер епохи	Точність моделі	Значення функції втрат	
			0	0.84116	0.3648605983018875	

Рисунок 4.3 – Фрагмент таблиці зі статистикою навчання та тестування моделей

Дана статистика включає в себе точність моделі, значення функції втрат при невірній класифікації та загальний час роботи моделі виражений у секундах.

Далі, на рисунках 4.4 та 4.5 можна побачити, відповідно, гістограму та графік ефективності алгоритмів.

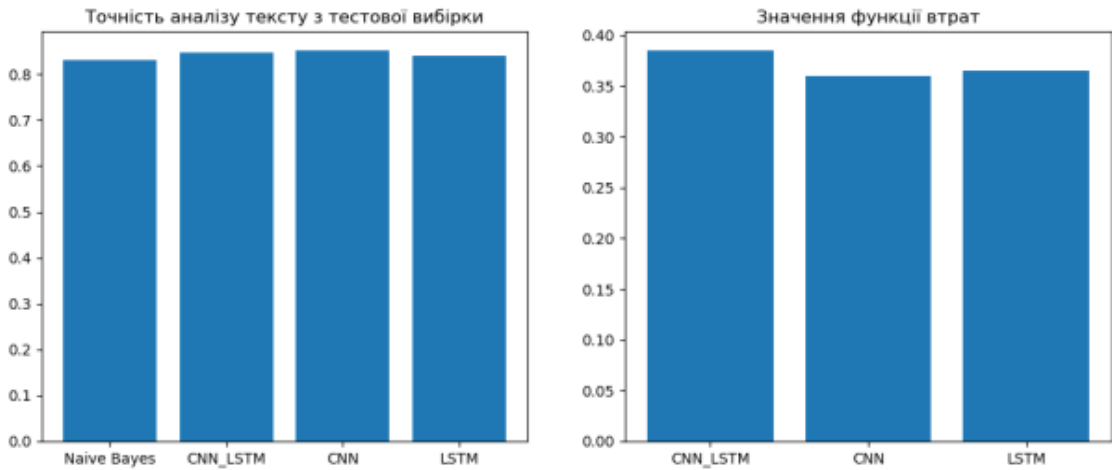


Рисунок 4.4 - Гістограма ефективності

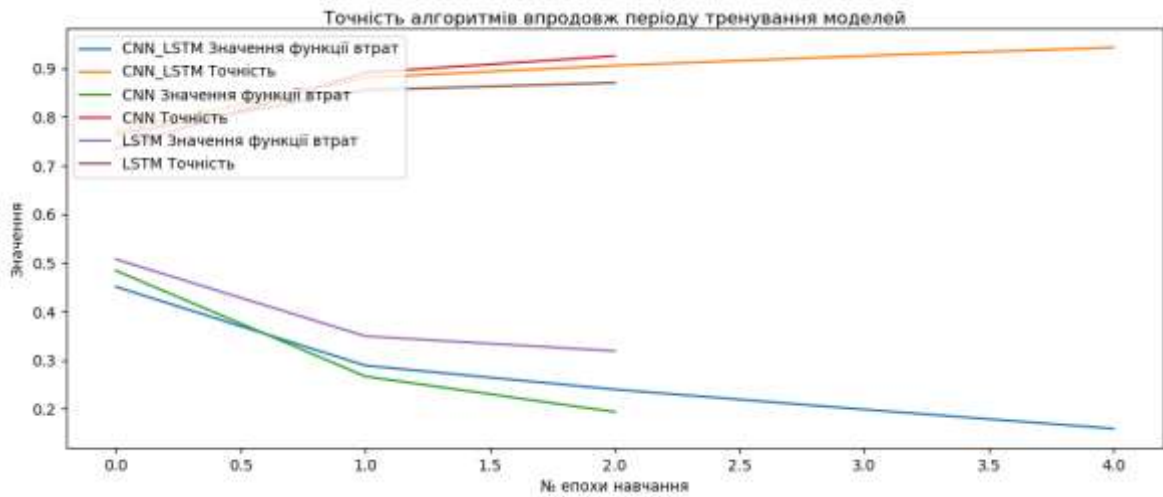


Рисунок 4.5 - Порівняльний графік точності моделей

### Висновок до розділу

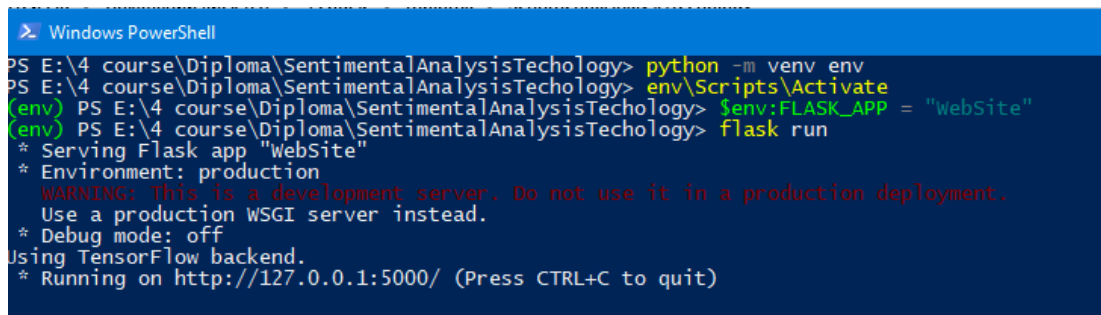
Отже, в даному розділі були описані основні засоби розробки даного програмного продукту, а також побудовані діаграми класів, компонентів та послідовності, що описують його структуру та послідовність взаємодії його частин.

Крім цього, було описано звіти, які формуються в результаті роботи програми.

## 5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

### 5.1 Керівництво користувача

Для того, щоб розпочати роботу з програмним продуктом спершу слід запустити локальний сервер. Тому, для його запуску слід ввести наступні команди у вікні Windows PowerShell, як показано на рисунку 5.1.



```
Windows PowerShell
PS E:\4 course\Diploma\SentimentalAnalysisTechnology> python -m venv env
PS E:\4 course\Diploma\SentimentalAnalysisTechnology> env\Scripts\Activate
(env) PS E:\4 course\Diploma\SentimentalAnalysisTechnology> $env:FLASK_APP = "webSite"
(env) PS E:\4 course\Diploma\SentimentalAnalysisTechnology> flask run
* Serving Flask app "webSite"
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
Using TensorFlow backend.
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Рисунок 5.1 – Запуск локального серверу за допомогою Windows PowerShell.

Після цього слід перейти за посиланням, яке з'явилося у вікні. Так, ми перейдемо до сторінки авторизації, яку ми можемо побачити на рисунку 5.2.

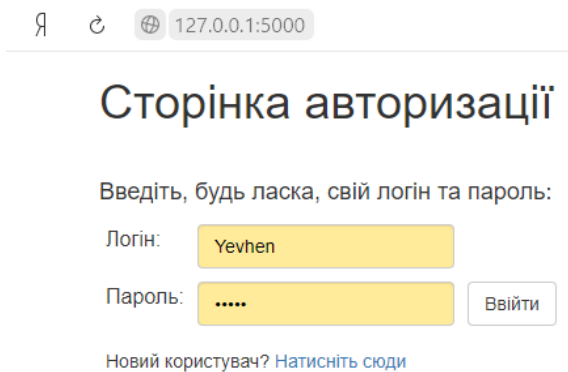


Рисунок 5.2 – Вигляд головної сторінки веб-сайту

Після цього користувач може увійти у систему за умови, якщо він вже зареєстрований або ж створити новий обліковий запис, натиснувши на посилання «Натисніть сюди».

В результаті, користувач переходить за посиланням на сторінку реєстрації, де йому слід задати власний придуманий логін та пароль. Для підтвердження паролю користувачу слід ввести його повторно у відповідне текстове поле. Вигляд даної процедури можна побачити на рисунку 5.3.

### Реєстрація нового користувача

Введіть логін:

Введіть пароль:

Повторіть пароль:

Рисунок 5.3 – Сторінка реєстрації

Після того, як всі поля форми було заповнено, користувач повинен натиснути на кнопку «Створити користувача», щоб зберегти свої дані у базі системи. Після створення облікового запису або при вході в систему під своїм логіном і паролем користувач переходить на сторінку, яка відображена на рисунку 5.4, із запропонованими йому моделями визначення тональності.

Виберіть модель визначення тональності тексту для її навчання

Список моделей:      Конфігурація моделі:      Список необхідних параметрів:

Найкращий Басов класифікатор

Нейронна мережа з ДКЧП      32:True,100.0,2.0,2:1,1      1. Довжина векторного представлення тексту.  
2. Масив для нульових елементів векторного представлення тексту.  
3. Кількість нейронів у шарі ДКЧП.

Комбінована нейронна мережа(ДКЧП+ЗНМ)      32:3,32:2,100,32,5,0,0      1. Довжина вхідних векторів з шару Embedding.  
2. Розмір ядра згорток.  
3. Кількість фільтрів для аналізу векторів слів.

Згорнута нейронна мережа (ЗНМ)      32:3-32:2,250,1,128,3      6. Кількість нейронів у другому звичайному шарі нейронної мережі.  
7. Розмір вибірки, для здійснення однієї ітерації тренування моделі.  
8. Кількість епох навчання нейронної мережі.

Кількість найпоширеніших слів у словнику:

Кількість слів у текстах:

Введіть текст для визначення його тональності:

Рисунок 5.4 – Веб-сторінка з моделями визначення тональності тексту



Виберіть одну з моделей для проведення аналізу текстів на їх тональність

Моделі Рекурентної нейронної мережі з ДКЧП :

Назва моделі	Кількість найчастіших слів	Довжина тексту	Конфігурація моделі	Назва файлу з моделлю
lstm_model	5000	100	[32, True, 100, 0.2, 0.2, 1, 128, 3]	5ec90a4650dd9443be84e9fe5

Моделі Згорткової нейронної мережі :

Назва моделі	Кількість найчастіших слів	Довжина тексту	Конфігурація моделі	Назва файлу з моделлю
snl_model	5000	100	[32, 3, 32, 2, 250, 1, 128, 3]	5ec909970dd9443be84e9fc8

Моделі Комбінованої нейронної мережі:

Назва моделі	Кількість найчастіших слів	Довжина тексту	Конфігурація моделі	Назва файлу з моделлю
snl_lstm_model	5000	100	[32, 3, 32, 2, 100, 32, 5, 0.05, 1, 1]	5ec905760dd9443be84e9fbb

Провести аналіз тексту

Рисунок 5.7 – Вигляд веб-сторінки з вибраними моделями

Результати роботи моделей визначення тональності тексту:

Введений користувачем текст: The movie was awesome. I love it

Назва моделі	Значення тональності	Визначений клас тексту
Рекурентна нейронна мережа з прямою короточасною пам'яттю (ДКЧП)	0.699	Позитивний
Згорткова нейронна мережа	0.407	Негативний
Комбінована нейронна мережа(ДКЧП+ЗНМ)	0.922	Позитивний

Рисунок 5.8 – Веб-сторінка з результатами класифікації тексту

Якщо користувач вирішив навчати моделі із заданою ним конфігурацією, то йому слід позначити моделі, які йому потрібно використати. Наприклад, якщо користувач позначив всі моделі, то сторінка з моделями буде виглядати наступним чином, як показано на рисунку 5.9.

Виберіть модель визначення тональності тексту для її навчання

Список моделей:      Конфігурація моделі:      Список необхідних параметрів:

- Негативний бажові класифікатор
- Нейронна мережа з ДКЧП      [32; True; 100; 0.2; 0.2; 1; 1]      1. Довжина векторного представлення тексту.  
2. Маска для нульових елементів векторного представлення тексту.  
3. Кількість нейронів у шарі ДКЧП.
- Комбінована нейронна мережа(ДКЧП+ЗНМ)      [32; 3; 32; 2; 100; 32; 5; 0.05]      1. Довжина вхідних векторів з шару Embedding.  
2. Розмір ядра згортки.  
3. Кількість фільтрів для аналізу векторів слів.
- Згорткова нейронна мережа (ЗНМ)      [32; 3; 32; 2; 250; 1; 128; 3]      1. Довжина вхідних векторів з шару Embedding.  
2. Розмір ядра згортки.  
3. Кількість фільтрів для аналізу векторів слів.

Кількість найпоширеніших слів у словнику: 5000

Кількість слів у текстах: 100

Введіть текст для визначення його тональності:

The movie was awesome. I love it

Рисунок 5.9 – Фрагмент веб-сторінки з позначеними моделями

Після того, як користувач натисне на кнопку «Почати тренування» вказані моделі будуть тренуватися на навчальній вибірці, а потім детальний звіт щодо статистики моделей та їх результати класифікації будуть відображені на веб-сторінці, яка показана на рисунках 5.10, 5.11, 5.12, 5.13 та 5.14 у вигляді таблиць, гістограм та графіку.

Результати роботи моделей визначення тональності тексту:

Введений користувачем текст: The movie was awesome. I love it

Назва моделі	Значення тональності	Визначений клас тексту
Найкращий Базовий Класифікатор	0.611	Позитивний
Комбінована нейронна мережа (ДКЧП+ЗНМ)	0.838	Позитивний
Згортоква нейронна мережа	0.387	Негативний
Рецурентна нейронна мережа з довгою короткочасною пам'яттю (ДКЧП)	0.681	Позитивний

Рисунок 5.10 – Результат класифікації введеного тексту

Назва моделі	Дані етапу			Час роботи	
Найкращий Базовий Класифікатор	Назва етапу	Статистика моделі		11.133193016052246	
	Тестування	Номер епохи	Точність моделі		
		0	0.83	Немає	
Комбінована нейронна мережа	Навчання	Статистика моделі		462.0311200618744	
		Номер епохи	Точність моделі		Значення функції втрат
		0	0.77044		0.4512212461962293
		1	0.88090		0.2691637852839665
		2	0.90588		0.23958501931142607
	3	0.92572	0.19873430901765829		
4	0.94358	0.15937635011195136			
	Тестування	Номер епохи	Точність моделі	Значення функції втрат	
		0	0.84656	0.364630063114624	

Рисунок 5.11 – Фрагмент таблиці зі статистикою навчання та тестування моделей

Згорюха нейронна мережа				21 682902812957784
Назва етапу	Статистика моделі			
Навчання	Номер епохи	Точність моделі	Значення функції втрат	
	0	0.7301E	0.4842801664113988	
	1	0.89128	0.26886857424003604	
	2	0.92556	0.19378882264900207	
Тестування	Номер епохи	Точність моделі	Значення функції втрат	
	0	0.85092	0.398198759508135	

Рекурентна нейронна мережа з ДКЧП				138 97327184677124
Назва етапу	Статистика моделі			
Навчання	Номер епохи	Точність моделі	Значення функції втрат	
	0	0.7638	0.5079958775920942	
	1	0.85337	0.3493305708129885	
	2	0.87048	0.3186530140390668	
Тестування	Номер епохи	Точність моделі	Значення функції втрат	
	0	0.84116	0.3648605983018875	

Рисунок 5.12 – Фрагмент таблиці зі статистикою навчання та тестування моделей

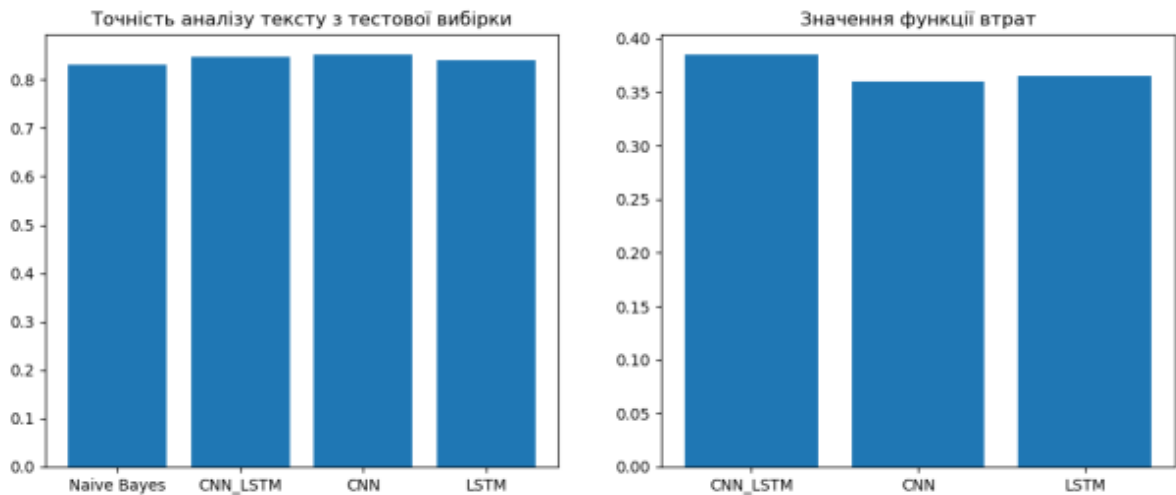


Рисунок 5.13 – Гістограми ефективності моделей на тестовій вибірці

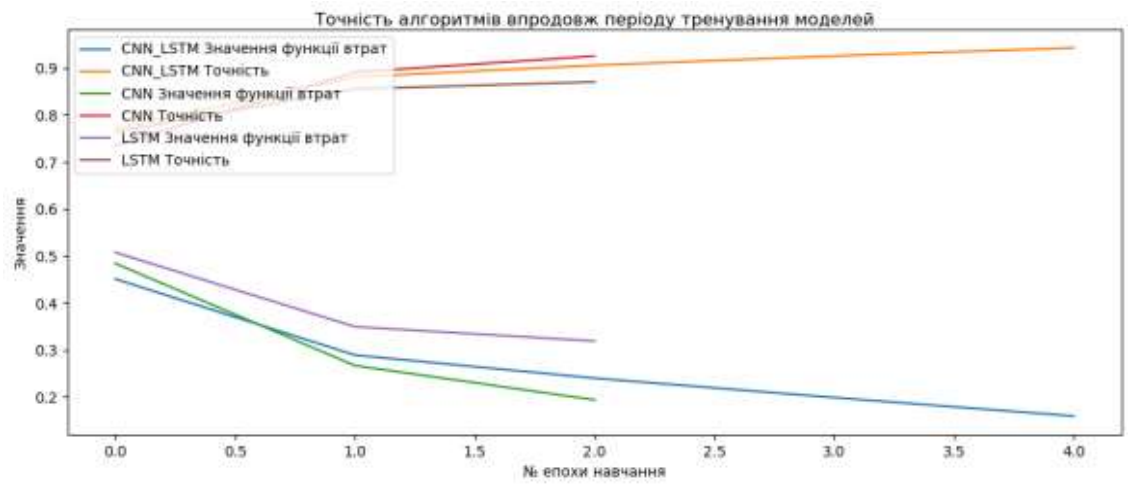


Рисунок 5.14 – Порівняльний графік збільшення точності та зниження значення функції втрат моделей нейронних мереж впродовж тренування

Управлінням доступу користувачів до інструментарію визначення тональності текстів керує адміністратор. Для того, щоб зайти на сторінку адміністратора необхідно вказати логін - “admin” і пароль - “admin”. А після цього, потрібно натиснути на кнопку «Ввійти». В такому випадку вигляд веб-сторінки можна представити на рисунку 5.15.

## Сторінка авторизації

Введіть, будь ласка, свій логін та пароль:

Логін:

Пароль:

Новий користувач? [Натисніть сюди](#)

Рисунок 5.15 – Веб-сторінка авторизації з введеними даними адміністратора

В результаті, після входу веб-сторінка набуває вигляду, який показаний на рисунку 5.16.

### Користувачі системи:

Номер користувача.№	Логін користувача	Доступ до моделей	Змінити доступ
5ec042ec804e6cd2c158400d	Yevhen	Так	<input type="button" value="Змінити доступ"/>
5ec90b0238518da602d7fe0a	Yura	Так	<input type="button" value="Змінити доступ"/>

Рисунок 5.16 – Веб-сторінка адміністратора системи

Адміністратор може змінити доступ певного користувача натиснувши на кнопку «Змінити доступ». Результат такої дії можна побачити на рисунку 5.17.

### Користувачі системи:

Номер користувача.№	Логін користувача	Доступ до моделей	Змінити доступ
5ec842ec8d4e5cd2c158480d	Yevhen	Ні	<input type="button" value="Змінити доступ"/>
5ec90b0238518da602d7fe0a	Yura	Так	<input type="button" value="Змінити доступ"/>

Рисунок 5.17 – Результат зміни доступу у користувача Yevhen

Таким чином, користувач Yevhen більше не зможе увійти у систему, щоб скористатися інструментарієм. Підтвердженням цього факту є рисунок 5.18.

### Сторінка авторизації

Введіть, будь ласка, свій логін та пароль:

Логін:

Пароль:

Новий користувач? [Натисніть сюди](#)

Помилка: У доступі до моделей відмовлено!

Рисунок 5.18 – Відмова в доступі користувача Yevhen до моделей

## 5.2 Випробування програмного продукту

### 5.2.1 Мета випробувань

Перевірка працездатності готового програмного продукту та тестування алгоритмів визначення тональності текстів на вибірці з вхідних даних і є метою проведення випробувань.

### 5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

### 5.2.3 Результати випробувань

В даному підрозділі ми опишемо результати експериментів стосовно розробленого програмного продукту. Хід проведення випробувань

					ДП 6303.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

складається з двох етапів:

- а) перевірка працездатності програмного продукту;
- б) перевірка реалізованих алгоритмів на різних вхідних даних.

Тому перейдемо до етапу а) та наведемо у таблиці 5.1 розроблені тестові сценарії.

Таблиця 5.1 – Тестові сценарії для перевірки працездатності програми

Перевірка функції надання доступу користувачам до моделей			
Передумови до виконання сценарію	Кроки виконання сценарію	Очікуваний результат	Результат перевірки
а) Локальний сервер запущений; б) Адміністратор перейшов за посиланням <a href="http://127.0.0.1:5000/">http://127.0.0.1:5000/</a> в) Адміністратор ввів логін “admin” та пароль “admin” у відповідні текстові поля та натиснув кнопку “Ввійти” г) У системі зареєстрований хоча б один користувач. д) У користувача доступ до моделей присутній	Вибрати користувача в таблиці користувачів	Користувач вибраний	Адміністратор вибрав користувача
	Натиснути на кнопку «Змінити доступ»	У користувача зник доступ до моделей	Адміністратор натиснув кнопку і доступ у вибраного користувача зник
Перевірка функції відкликання доступу у користувачів до моделей			
Передумови до виконання сценарію	Кроки виконання сценарію	Очікуваний результат	Результат перевірки
а) Передумови а)-г) з попереднього тестового сценарію виконані б) У користувача доступу до моделей немає	Вибрати користувача в таблиці користувачів	Користувач вибраний	Адміністратор вибрав користувача
	Натиснути на кнопку «Змінити доступ»	У користувача з’явився доступ до моделей	Адміністратор натиснув кнопку і доступ у вибраного користувача з’явився

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 5.1

Перевірка функції задання власного тексту			
Передумови до виконання сценарію	Кроки виконання сценарію	Очікуваний результат	Результат перевірки
а) Користувач перейшов на сторінку з моделями визначення тональності	Ввести текст для аналізу у відповідне текстове поле	Текст введений	Користувач ввів текст «The movie was awesome. I love it»
	Натиснути на кнопку «Почати навчання».	Текст введений вірно: повідомлення про помилку немає	Текст введений вірно : повідомлення про помилку немає
Перевірка функції отримання доступу до АРІ системи та до його веб-сайту			
Передумови до виконання сценарію	Кроки виконання сценарію	Очікуваний результат	Результат перевірки
а) Локальний сервер запущений; б) Користувач перейшов за посиланням <a href="http://127.0.0.1:5000/">http://127.0.0.1:5000/</a> ; в) Користувач не має облікового запису	Натиснути на посилання «Натисніть сюди»	Посилання натиснуте і користувач переходить на сторінку реєстрації	Користувач перейшов на сторінку реєстрації
	Ввести логін, пароль і повтор паролю у відповідні текстові поля	Логін, пароль, повтор паролю введені	Користувач ввів логін “Yevhen”, пароль “12345” і повтор паролю “12345”
	Натиснути на кнопку «Створити користувача»	Обліковий запис створено та користувач перейшов на сторінку з моделями визначення тональності	Користувач зареєструвався у системі та перейшов на сторінку з моделями визначення тональності

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 5.1

Перевірка функції вибору користувачем моделей визначення тональності			
Передумови до виконання сценарію	Кроки виконання сценарію	Очікуваний результат	Результат перевірки
<p>а) Користувач перейшов на сторінку з моделями визначення тональності</p> <p>б) Користувач вирішив запустити навчання моделей</p>	Позначити галками моделі визначення тональності	Користувач позначив галками моделі	Користувач позначив модель найного баєсового класифікатора
<p>а) Користувач перейшов на сторінку з моделями визначення тональності</p> <p>б) Користувач вирішив завантажити моделі з бази даних</p> <p>в) В базі даних присутня хоча б одна модель визначення тексту</p>	Ввести текст для аналізу в текстове поле	Користувач ввів текст	Користувач ввів текст «The movie was awesome. I love it»
	Натиснути на кнопку «Завантажити дані з бази даних»	Користувач переходить на нову веб-сторінку з таблицями завантажених моделей	Користувач перейшов на нову веб-сторінку з трьома таблицями з даними про моделі
	Натиснути на кнопку «Вибрати модель»	У списку моделей залишиться одна модель	У списку залишилася одна модель

Продовження таблиці 5.1

Перевірка функції завантаження моделей та їх конфігурації з бази даних			
Передумови до виконання сценарію	Кроки виконання сценарію	Очікуваний результат	Результат перевірки
а) Користувач перейшов на сторінку з моделями визначення тональності б) Користувач вирішив завантажити моделі з бази даних в) В базі даних присутня хоча б одна модель визначення тексту	Ввести текст для аналізу в текстове поле	Користувач ввів текст	Користувач ввів текст «The movie was awesome. I love it»
	Натиснути на кнопку «Завантажити дані з бази даних»	Користувач переходить на нову веб-сторінку з таблицями завантажених моделей	Користувач перейшов на нову веб-сторінку з трьома таблицями з даними про моделі

Перевірка функції встановлення параметрів конфігурації

Передумови до виконання сценарію	Кроки виконання сценарію	Очікуваний результат	Результат перевірки
а) Користувач перейшов на сторінку з моделями визначення тональності	Позначити галкою одну з моделей нейронних мереж	Поле введення конфігурації розблокувалося	Користувач вибрав згорткову нейронну мережу – її текстове поле розблокувалося
	Ввести текст для конфігурації у відповідне текстове поле моделі	Конфігурація введена	Користувач ввів конфігурацію «32;3;32;2;250;1;128;3»
	Натиснути на кнопку «Почати навчання».	Конфігурація введена вірно: повідомлення про невірний формат даних не спливає	Конфігурація введена вірно : повідомлення про помилку немає

Змн.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 5.1

Перевірка функції запуску тренування моделей			
Передумови до виконання сценарію	Кроки виконання сценарію	Очікуваний результат	Результат перевірки
а) Користувач перейшов на сторінку з моделями визначення тональності б) Користувач вибрав моделі в) Користувач ввів конфігурації моделей, для яких потрібно г) Користувач задав текст д) База даних з навчальною та тестовою вибірками присутня	Натиснути на кнопку «Почати навчання»	Тренування моделей почалося	Користувач натиснув на кнопку «Почати навчання» і навчання моделей успішно почалося
Перевірка функції перегляду звіту з результатами роботи алгоритмів та їх ефективності			
Передумови до виконання сценарію	Кроки виконання сценарію	Очікуваний результат	Результат перевірки
а) Користувач успішно запустив навчання хоча б однієї моделі визначення тональності тексту б) Користувач задав текст	Дочекатися закінчення навчання моделей	Тренування моделей завершилося і з'явилася веб-сторінка зі звітом	Після запуску навчання моделей через деякий проміжок часу з'явилася веб-сторінка з детальною інформацією про результати навчання алгоритмів та класифікації заданого тексту

Перейдемо до етапу б), де ми перевіримо ефективність розроблених алгоритмів на тестовій вибірці. Для цього був сформований масив даних, що складається з 25 000 відгуків глядачів фільмів, в якому містяться, як позитивні, так і негативні відгуки. Також, для порівняння нейронних мереж між собою емпіричним шляхом було визначено наступний набір конфігурацій, при якому точність даних моделей в ході тренування є приблизно однаковою:

- а) кількість найбільш поширених слів у словнику – 5000;
- б) довжина тексту – 100 слів;
- в) конфігурація рекурентної нейронної мережі:  
«32;True;100;0.2;0.2;1;128;3»;
- г) конфігурація комбінованої нейронної мережі:  
«32;3;32;2;100;32;5;0.05;1;1»;
- д) конфігурація згорткової нейронної мережі:  
«32;3;32;2;250;1;128;3».

Таким чином, це дозволить нам оцінити ефективність нейронних мереж під час тестування за однакових умов.

Результати даної перевірки можна побачити на рисунку 5.18.

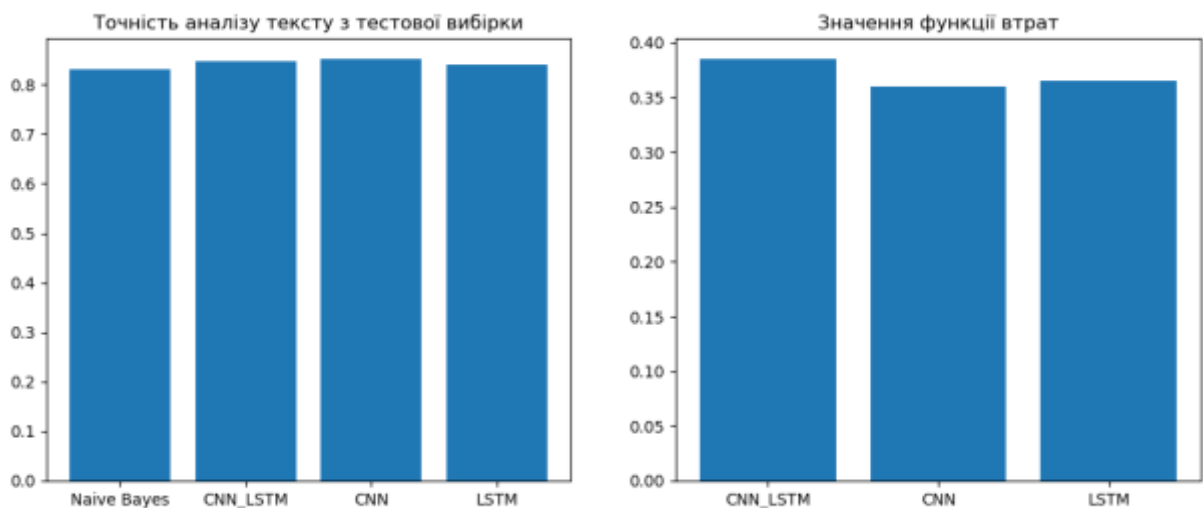


Рисунок 5.18 – Гістограми ефективності моделей на тестовій вибірці

На гістограмі точності аналізу тексту з тестової вибірки можна помітити, що точність аналізу нейронними мережами переважає точність аналізу найвним класифікатором Баеса. Також, можна прийти до висновку, що точність алгоритму згорткової нейронної мережі, яка позначається як «CNN», є найвищою. Цей факт також приводиться даними з сусідньої гістограми, в якій вказано значення функції втрат при невірній класифікації текстів нейронною мережею. Таким чином, на цій гістограмі можна чітко побачити, що дана модель помилялася найменше серед своїх аналогів.

З іншого боку, комбінована нейронна мережа, яка позначається як «CNN\_LSTM» має найбільше значення функції втрат. Хоча, вона посідає друге місце по точності серед всіх алгоритмів, що показує її досить непогану ефективність.

### Висновок до розділу

Отже, в даному розділі було наведено керівництво користувача, в якому була описана послідовність дій, які повинен виконувати користувач програмного продукту при роботі з ним. Крім цього, були додані фрагменти інтерфейсу програми.

Далі були описані тестові сценарії, в яких було більш детально розглянуто передумови до виконання певних функцій програми, а також які результати отримуються в ході виконання програми.

Врешті-решт, було продемонстровано ефективність алгоритмів визначення тональності текстів на тестовій вибірці, яка містить 25000 відгуків, і наведено конфігурації моделей, при яких були досягнуті вказані результати.

Таким чином, найкращою у класифікації продемонструвала себе модель згорткової нейронної мережі, а найгіршим виявився наївний баєсів класифікатор.

					ДП 6303.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

## ЗАГАЛЬНІ ВИСНОВКИ

У першому розділі загальних положень було описано предметне середовище інформаційної технології визначення тональності тексту та процес діяльності цієї системи. Було встановлено, що аналіз тональності тексту є досить потужним інструментом для визначення емоційної складової відгуку покупця щодо придбаного товару. Ця технологія дозволяє власникам великих компаній отримувати цінну інформацію з соціальних мереж щодо становища їх продукції на ринку. А також, я розглянув існуючі аналоги мого програмного продукту та визначив їх ключові особливості.

В другому розділі був визначений перелік вхідних та вихідних даних. Так, основними параметрами, які подаються на вхід є конфігурація моделей визначення тональності тексту, навчальна та тренувальна вибірки даних із проставленими відмітками, що вказують на клас тексту, максимальна кількість слів та число найпоширеніших слів у текстах навчальної та тестової вибірки. До вихідних даних належать результати щодо класифікації вхідних текстів та точність визначення цих класів, які представлені у вигляді у таблиць, гістограм та графіків. Конфігурація моделей та самі моделі визначення тональності зберігаються в базі даних разом з даними про облікові записи користувачів.

В третьому розділі були наведені змістовна та математичні постановки задачі визначення тональності тексту. Також, було зазначено які функції повинен виконувати програмний продукт для вирішення поставленої задачі. Далі, було встановлено та описано основні методи виконання даної задачі.

В четвертому розділі були описані основні засоби розробки даного програмного продукту, а також побудовані діаграми класів, компонентів та послідовності, що описують його структуру та послідовність взаємодії його частин. Крім цього, було описано звіти, які формуються в результаті роботи програми.

					ДП 6303.00.000 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

У п'ятому розділі було наведено керівництво користувача, в якому була описана послідовність дій, які повинен виконувати користувач програмного продукту при роботі з ним. Крім цього, були додані фрагменти інтерфейсу програми. Далі були описані тестові сценарії, в яких було більш детально розглянуто передумови до виконання певних функцій програми, а також які результати отримуються в ході виконання програми. Врешті-решт, було продемонстровано ефективність алгоритмів визначення тональності текстів на тестовій вибірці, яка містить 25000 відгуків і наведено конфігурації моделей, при яких були досягнуті вказані результати.

Таким чином, найкращою у класифікації продемонструвала себе модель згорткової нейронної мережі, а найгіршим виявився найвний баєсів класифікатор.

					ДП 6303.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

**ПЕРЕЛІК ПОСИЛАНЬ**

1. Блінков Є.М. Технології аналізу тональності текстів // Матеріали ІV всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2020) – м. Київ.: НТУУ «КПІ ім. Ігоря Сікорського», 24-30 квітня 2020 р. – С. 28-31.
2. About Keras [Електронний ресурс] – Режим доступу до ресурсу: <https://keras.io/about/>.
3. An LSTM Approach to Short Text Sentiment Classification with Word Embeddings [Електронний ресурс] / W.Jenq-Haur, L. Ting-Wei, L. Xiong, W. Long. – 2018. – Режим доступу до ресурсу: <https://www.aclweb.org/anthology/O18-1021.pdf>
4. Brownlee J. How to Save and Load Your Keras Deep Learning Model [Електронний ресурс] / Jason Brownlee. – 2019. – Режим доступу до ресурсу: <https://machinelearningmastery.com/save-load-keras-deep-learning-models/>.
5. Fontanella C. The Best 8 Sentiment Analysis Tools in 2020 [Електронний ресурс] / Clint Fontanella. – 2020. – Режим доступу до ресурсу: <https://blog.hubspot.com/service/sentiment-analysis-tools>.
6. Grinvald B. WHAT IS SENTIMENT ANALYSIS? [Електронний ресурс] / Boaz Grinvald. – 2019. – Режим доступу до ресурсу: <https://www.revuze.it/blog/what-is-sentiment-analysis/>.
7. Gupta S. Sentiment Analysis: Concept, Analysis and Applications [Електронний ресурс] / Shashank Gupta. – 2018. – Режим доступу до ресурсу: <https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17>.
8. IMDB movie review sentiment classification dataset [Електронний ресурс] – Режим доступу до ресурсу: <https://keras.io/api/datasets/imdb/>.

9. Matplotlib: Visualization with Python [Электронный ресурс] – Режим доступа до ресурсу: <https://matplotlib.org/>.
10. McCaffrey J. Saving and Displaying Keras Model Weights [Электронный ресурс] / James McCaffrey. – 2018. – Режим доступа до ресурсу: <https://jamesmccaffrey.wordpress.com/2018/02/15/saving-and-displaying-keras-model-weights/>.
11. MongoDB - GridFS [Электронный ресурс] – Режим доступа до ресурсу: [https://www.tutorialspoint.com/mongodb/mongodb\\_gridfs.htm](https://www.tutorialspoint.com/mongodb/mongodb_gridfs.htm).
12. NumPy in Python | Set 1 (Introduction) [Электронный ресурс] – Режим доступа до ресурсу: <https://www.geeksforgeeks.org/numpy-in-python-set-1-introduction/>.
13. Rouse M. MongoDB [Электронный ресурс] / Margaret Rouse. – 2018. – Режим доступа до ресурсу: <https://searchdatamanagement.techtarget.com/definition/MongoDB>.
14. Sentiment Analysis Explained [Электронный ресурс] – Режим доступа до ресурсу: <https://www.lexalytics.com/technology/sentiment-analysis>.
15. Smetanin S. Sentiment Analysis of Tweets using Multinomial Naive Bayes [Электронный ресурс] / Sergey Smetanin. – 2018. – Режим доступа до ресурсу: <https://towardsdatascience.com/sentiment-analysis-of-tweets-using-multinomial-naive-bayes-1009ed24276b>.
16. What is Python? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.pythonforbeginners.com/learn-python/what-is-python/>.
17. What is Flask [Электронный ресурс] – Режим доступа до ресурсу: <https://dev.to/amigosmaker/what-is-flask-1d63>.

Додаток А

**Тексти програмного коду**  
**Інформаційна технологія визначення тональності текстів**

---

(Найменування програми (документа))

*DVD-R*

---

(Вид носія даних)

*17 арк, 51 Кб*

---

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2020 року

					ДП 6303.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66



```

@app.route('/AnalysisPage')
def index():
    return render_template('AnalysisPage.html')
    # return "Hello World"

@app.after_request
def add_header(response):
    response.headers['Cache-Control'] = 'public, max-age=0'
    return response

def getValueFromDictionary(argument, dictionary):
    value = None
    try:
        value = dictionary.get(argument, "")
    except:
        pass
    return value

def getConfiguration(userInputConfig, params_number):
    params = []
    params = userInputConfig.split(";", params_number)
    return params[:params_number]

@app.route('/load', methods=['POST'])
def loadData():
    global text
    userInputData = request.form.to_dict()
    text = getValueFromDictionary('inputText', userInputData)
    global LSTM_Table
    global CNN_LSTM_Table
    global CNN_Table
    lstmInput, LSTM_Table = loadConfiguration("lstm_model")
    dataTable.update({"LSTM": LSTM_Table})
    cnnInput, CNN_Table = loadConfiguration("cnn_model")
    dataTable.update({"CNN": CNN_Table})
    cnn_lstmInput, CNN_LSTM_Table = loadConfiguration("cnn_lstm_model")
    dataTable.update({"CNN_LSTM": CNN_LSTM_Table})
    LSTM_Table.border = True
    return render_template('dataTable.html', LSTM_table=LSTM_Table, CNN_table=CNN_Table, CNN_LSTM_table=CNN_LSTM_Table)

@app.route('/cell/<string: id>', methods=['GET', 'POST'])
def choose(id):
    global LSTM_Table
    global CNN_LSTM_Table
    global CNN_Table
    record = findRecordById(id)
    recordTable = makeModelTable(record)
    modelType = record[0]['ModelName']
    if modelType == "lstm_model":
        LSTM_Table = recordTable
        chosenModels.update({'LSTM': record[0]})
    if modelType == "cnn_model":
        CNN_Table = recordTable
        chosenModels.update({'CNN': record[0]})
    if modelType == "cnn_lstm_model":
        CNN_LSTM_Table = recordTable
        chosenModels.update({'CNN_LSTM': record[0]})
    print(record)
    return render_template('dataTable.html', LSTM_table=LSTM_Table, CNN_table=CNN_Table, CNN_LSTM_table=CNN_LSTM_Table)

@app.route('/run', methods=['POST'])
def runModels():
    estimator = Estimator()
    filenames = []
    review_len_s = []
    filenames.append(getValueFromDictionary('ModelFileName', getValueFromDictionary('LSTM', chosenModels)))
    filenames.append(getValueFromDictionary('ModelFileName', getValueFromDictionary('CNN', chosenModels)))
    filenames.append(getValueFromDictionary('ModelFileName', getValueFromDictionary('CNN_LSTM', chosenModels)))
    review_len_s.append(getValueFromDictionary('MaxReviewLen', getValueFromDictionary('LSTM', chosenModels)))
    review_len_s.append(getValueFromDictionary('MaxReviewLen', getValueFromDictionary('CNN', chosenModels)))
    review_len_s.append(getValueFromDictionary('MaxReviewLen', getValueFromDictionary('CNN_LSTM', chosenModels)))
    results, tableResult = estimator.runAll(filenames, review_len_s, text)
    print(results)
    table = makeResultTable(tableResult)
    return render_template("result.html", text=text, table=table)

@app.route('/result', methods=['POST'])
def result():
    if request.method == "POST":
        userInputData = request.form.to_dict()
        Bayes_Input = getValueFromDictionary('Bayes_Input', userInputData)
        LSTM_Input = getValueFromDictionary('LSTM_Input', userInputData)
        combinedInput = getValueFromDictionary('CombinedInput', userInputData)
        CNN_Input = getValueFromDictionary('CNN_Input', userInputData)

        top_words = int(getValueFromDictionary('frequency', userInputData))
        review_max_length = int(getValueFromDictionary('length', userInputData))
        userText = getValueFromDictionary('inputText', userInputData)

        print(userInputData)
        print(LSTM_Input, combinedInput, CNN_Input, top_words, review_max_length, userText)
        # numbers_to_strings(argument)
        totalConfig = [LSTM_Input, combinedInput, CNN_Input, top_words, review_max_length]

        LSTM_config = getConfiguration(LSTM_Input, 8)
        CNN_config = getConfiguration(CNN_Input, 8)

```

						ДП 6303.00.000 ПЗ	Арк. 68
Змн.	Арк.	№ докум.	Підпис	Дата			



```

predictions.append(bayesPrediction)
text_results[0] = bayesPrediction
textTones[0] = "{:.3f}".format(classTone)
definedClasses[0] = definedClass
if len(CNN_LSTM_Config)!=0:
    start_time = time.time()

    historyCNN_LSTM, evalCNN_LSTM,cnn_lstmPrediction,classTone, definedClass = self.estimateCNN_LSTM(top_words, review_len, CNN_LSTM_Config, userText,test_x, test_y,
train_x, train_y )
    start_time = -start_time + time.time()
    time_results[1] = start_time
    histories[0]=tuple(("CNN_LSTM", historyCNN_LSTM))
    lossList.update({"CNN_LSTM": evalCNN_LSTM[0]})
    accList.update( {"CNN_LSTM": evalCNN_LSTM[1]})
    predictions.append(cnn_lstmPrediction)
    text_results[1] = cnn_lstmPrediction
    textTones[1] = "{:.3f}".format(classTone)
    definedClasses[1] = definedClass

if len(CNN_Config) != 0:
    start_time = time.time()
    historyCNN, evalCNN,cnnPrediction,classTone, definedClass = self.estimateCNN(top_words, review_len, CNN_Config, userText,test_x, test_y, train_x, train_y )
    start_time = -start_time + time.time()
    time_results[2] = start_time
    histories[1]=tuple(("CNN", historyCNN))
    lossList.update({"CNN": evalCNN[0]})
    accList.update( {"CNN": evalCNN[1]})
    predictions.append(cnnPrediction)
    text_results[2] = cnnPrediction
    textTones[2] = "{:.3f}".format(classTone)
    definedClasses[2] = definedClass
if len(LSTM_Config) != 0:
    start_time = time.time()

    historyLSTM, evalLSTM,LSTM_Prediction,classTone, definedClass = self.estimateLSTM(top_words, review_len, LSTM_Config, userText,test_x, test_y, train_x, train_y )
    start_time = -start_time + time.time()
    time_results[3] = start_time
    histories[2]=tuple(("LSTM", historyLSTM))
    lossList.update({"LSTM": evalLSTM[0]})
    accList.update( {"LSTM": evalLSTM[1]})
    predictions.append(LSTM_Prediction)
    text_results[3] = LSTM_Prediction
    textTones[3] = "{:.3f}".format(classTone)
    definedClasses[3] = definedClass
tableResult = [dict(Text=text_results[0], Prediction=textTones[0], PredictionClass=definedClasses[0]),
                dict(Text=text_results[1], Prediction=textTones[1], PredictionClass=definedClasses[1]),
                dict(Text=text_results[2], Prediction=textTones[2], PredictionClass=definedClasses[2]),
                dict(Text=text_results[3], Prediction=textTones[3], PredictionClass=definedClasses[3])]
plot = self.outputGraph(histories)
self.showTestResultsGraph(accList, lossList, plot)

print(tableResult)
report = self.makeReport(histories, accList, lossList, time_results)
return histories,lossList,accList,predictions,text_results,tableResult,report

def makeReport(self,histories,accList,lossList,timeResults):
    report=[dict(ModelName="Найвний Басвів Класифікатор", StageData=
    [dict(StageName="Тестування",subStatisticsTable=
    [dict( EpochNumber=0,Accuracy= getValueFromDictionary('Naive Bayes', accList),Loss="N/A")
    ],TimeCol=timeResults[0]),
    dict(ModelName="Комбінована нейронна мережа", StageData=
    [dict(StageName="Навчання", subStatisticsTable=self.getStatisticsList(histories, 0)),
    dict(StageName="Тестування", subStatisticsTable=
    [dict(EpochNumber=0, Accuracy=getValueFromDictionary('CNN_LSTM', accList), Loss=getValueFromDictionary("CNN_LSTM",lossList))]
    ), TimeCol=timeResults[1]),
    dict(ModelName="Згорькова нейронна мережа", StageData=
    [dict(StageName="Навчання", subStatisticsTable=self.getStatisticsList(histories, 1)),
    dict(StageName="Тестування", subStatisticsTable=
    [dict(EpochNumber=0, Accuracy=getValueFromDictionary('CNN',accList), Loss=getValueFromDictionary("CNN",lossList))]
    ),TimeCol=timeResults[2]),
    dict(ModelName="Рекурентна нейронна мережа з ДКЧП", StageData=
    [dict(StageName="Навчання", subStatisticsTable=self.getStatisticsList(histories, 2)),
    dict(StageName="Тестування", subStatisticsTable=
    [dict(EpochNumber=0, Accuracy=getValueFromDictionary('LSTM',accList), Loss=getValueFromDictionary("LSTM",lossList))]
    ), TimeCol=timeResults[3])
    ]
    ]
    return report
def getStatisticsList(self,histories,algoNumber):
    statList=[]
    if histories[algoNumber]!='':
        for i in range(0,len(histories[algoNumber][1].history['acc'])):
            statList.append(dict(EpochNumber=i, Accuracy=histories[algoNumber][1].history['acc'][i], Loss=histories[algoNumber][1].history['loss'][i]))
    return statList

def showTestResultsGraph(self, accList, lossList, plot):
    accNames = list(accList.keys())
    accValues = list(accList.values())
    lossNames = list(lossList.keys())
    lossValues = list(lossList.values())
    plot.figure(1)
    fig, axs = plot.subplots(1, 2, figsize=(13, 5))
    axs[0].bar(accNames, accValues)
    axs[0].set_title("Точність аналізу тексту з тестової вибірки")
    axs[1].bar(lossNames, lossValues)
    axs[1].set_title("Значення функцій втрат")
    fig.suptitle("Точність моделей на текстах з тестової вибірки", y=0.99)
    plot.savefig("/static/graphs/graph1.png")

def outputGraph(self, histories):
    plt.figure(figsize=(13, 5))
    for history in histories:

```

```

if history != "":
    algoName, algoHistory = history
    plt.plot(algoHistory.history['loss'], label=('%%s Значення функцій втрат' % algoName))
    plt.plot(algoHistory.history['acc'], label=('%%s Точність' % algoName))
    plt.legend(loc="upper left")
plt.xlabel('№ епохи навчання')
plt.ylabel('Значення')
plt.title('Точність алгоритмів впродовж періоду тренування моделей', y=0.99)
plt.savefig("./static/graphs/graph2.png")
return plt

def estimateLSTM(self, top_words, review_len, LSTM_Config, userText, test_x, test_y, train_x, train_y):
    config = LSTMConfiguration(LSTM_Config[0],LSTM_Config[1],LSTM_Config[2],LSTM_Config[3],
        LSTM_Config[4],LSTM_Config[5],LSTM_Config[6],LSTM_Config[7])

    lstm = LSTM(top_words, review_len, config)

    model, history, eval_epoch_history = lstm.defineModel(test_x, test_y, train_x, train_y)

    lstm_result, prediction, definedClass = lstm.runModel(model, userText, review_len)

    return history, eval_epoch_history, lstm_result, prediction, definedClass

def estimateCNN(self, top_words, review_len, CNN_Config, userText, test_x, test_y, train_x, train_y):
    config=CNNConfiguration(CNN_Config[0],CNN_Config[1],CNN_Config[2],CNN_Config[3],
        CNN_Config[4],CNN_Config[5],CNN_Config[6],CNN_Config[7])

    cnn=CNN(top_words, review_len, config)

    model, history, eval_epoch_history = cnn.defineModel(test_x, test_y, train_x, train_y)

    cnn_result, prediction, definedClass =cnn.runModel(model, userText, review_len)

    return history, eval_epoch_history, cnn_result, prediction, definedClass

def estimateCNN_LSTM(self, top_words, review_len, CNN_LSTM_Config, userText, test_x, test_y, train_x, train_y):
    config=CNN_LSTMConfiguration(CNN_LSTM_Config[0],CNN_LSTM_Config[1],CNN_LSTM_Config[2],CNN_LSTM_Config[3],CNN_LSTM_Config[4],
        CNN_LSTM_Config[5],CNN_LSTM_Config[6],CNN_LSTM_Config[7],CNN_LSTM_Config[8],CNN_LSTM_Config[9])

    cnn_lstm = CNN_LSTM(top_words, review_len, config)

    model, history, eval_epoch_history = cnn_lstm.defineModel(test_x, test_y, train_x, train_y)

    cnn_lstm_result, prediction, definedClass = cnn_lstm.runModel(model, userText, review_len)

    return history, eval_epoch_history, cnn_lstm_result, prediction, definedClass

def estimateBayes(self, top_words, review_len, userText, training_set, training_labels, validation_set, validation_labels):
    print(training_set[0], training_labels[0])
    print(validation_set[0], validation_labels[0])
    NBClassifier = NaiveBayesClassifier()
    model, train_results = NBClassifier.defineModel(validation_set, validation_labels, training_set,
        training_labels)
    results, acc, bayes_result, class Tone, definedClass=NBClassifier.runModel(model, userText, None)
    return train_results, results, acc, bayes_result, class Tone, definedClass

def runCNN(self, filename, review_len, userText):
    cnn = CNN()
    model = cnn.loadModel(filename)
    cnn_result, prediction, definedClass = cnn.runModel(model, userText, review_len)
    return cnn_result, prediction, definedClass

def runCNN_LSTM(self, filename, review_len, userText):
    cnn_lstm = CNN_LSTM()
    model = cnn_lstm.loadModel(filename)
    cnn_lstm_result, prediction, definedClass = cnn_lstm.runModel(model, userText, review_len)
    return cnn_lstm_result, prediction, definedClass

def runLSTM(self, filename, review_len, userText):
    lstm = LSTM()
    model = lstm.loadModel(filename)
    lstm_result, prediction, definedClass = lstm.runModel(model, userText, review_len)
    return lstm_result, prediction, definedClass

def runAll(self, filenames, review_len_s, userText):
    results=["", "", ""]
    class Tones=["", "", ""]
    definedClasses=["", "", ""]
    if filenames[0]!=None:
        modelName, classTone, definedClass=self.runLSTM(filenames[0], review_len_s[0], userText)
        results[0]=modelName
        classTones[0] = "{:3f}".format(classTone)
        definedClasses[0] = definedClass
    if filenames[1]!=None:
        modelName, classTone, definedClass=self.runCNN(filenames[1], review_len_s[1], userText)
        results[1]=modelName
        classTones[1] = "{:3f}".format(classTone)
        definedClasses[1] = definedClass
    if filenames[2]!=None:
        modelName, classTone, definedClass=self.runCNN_LSTM(filenames[2], review_len_s[2], userText)
        results[2]=modelName
        classTones[2] = "{:3f}".format(classTone)
        definedClasses[2] = definedClass

tableResult=[dict(Text=results[0], Prediction=classTones[0], PredictionClass=definedClasses[0]),dict(Text=results[1], Prediction=classTones[1], PredictionClass=definedClasses[1]),dict(Text=results[2], Prediction=classTones[2], PredictionClass=definedClasses[2])]
print(tableResult)
return results, tableResult

def deleteSavedImages():
    print("Removing old graphs... \n")
    mp = ".\static\graphs\"
    for f in os.listdir(mp):
        os.remove(os.path.join(mp, f))

def getValueFromDictionary(argument, dictionary):
    value = None
    try:
        value = dictionary.get(argument, "")
    
```

```
except:
    pass
return value
```

### Файл DataTable.py

```
from flask_table import Table, Col, ButtonCol, NestedTableCol

class DataTable(Table):
    classes = ['table', 'table-striped', 'table-condensed', 'table-hover']
    _id = Col('Id', show=False)
    ModelName = Col('Назва моделі')
    TopWords = Col('Кількість найчастіших слів')
    MaxReviewLen = Col('Довжина тексту')
    Configuration = Col('Конфігурація моделі')
    ModelFileName = Col('Назва файлу з моделлю')
    edit = ButtonCol('Вибрати модель', 'choose', url_kwargs=dict(id='_id'))

class ModelTable(Table):
    classes = ['table', 'table-striped', 'table-condensed', 'table-hover']
    _id = Col('Id', show=False)
    ModelName = Col('Назва моделі')
    TopWords = Col('Кількість найчастіших слів')
    MaxReviewLen = Col('Довжина тексту')
    Configuration = Col('Конфігурація моделі')
    ModelFileName = Col('Назва файлу з моделлю')

class ResultTable(Table):
    classes = ['table', 'table-striped', 'table-condensed', 'table-hover']
    Text = Col('Назва моделі')
    Prediction = Col('Значення тональності')
    PredictionClass = Col('Визначений клас тексту')

class StatisticsTable(Table):
    classes = ['table', 'table-striped', 'table-condensed', 'table-hover']
    EpochNumber = Col('Номер епохи')
    Accuracy = Col('Точність моделі')
    Loss = Col('Значення функції втраг')

class SubStageTable(Table):
    classes = ['table', 'table-striped', 'table-condensed', 'table-hover']
    StageName = Col('Назва етапу')
    subStatisticsTable = NestedTableCol('Статистика моделі', StatisticsTable)

class ReportTable(Table):
    classes = ['table', 'table-striped', 'table-condensed', 'table-hover']
    ModelName = Col('Назва моделі')
    StageData = NestedTableCol('Дані етапу', SubStageTable)
    TimeCol = Col('Час роботи')
```

### Файл DataLoader.py

```
import json
import string

import tensorflow.keras as K
import numpy as np

def loadData(top_words, max_review_len):
    print("Loading data, max unique words = %d words\n" % top_words)
    (train_x, train_y), (test_x, test_y) = \
        K.datasets.imdb.load_data(seed=1, num_words=top_words)
    train_x = K.preprocessing.sequence.pad_sequences(train_x, truncating='pre', padding='pre', maxlen=max_review_len) # pad and chop!
    test_x = K.preprocessing.sequence.pad_sequences(test_x, truncating='pre', padding='pre', maxlen=max_review_len)

    return test_x, test_y, train_x, train_y

def split_review_data(reviews, sentiment_numerical_val, split=900, remove_punc=True, separation=""):
    training_set = []
    training_labels = []
    validation_set = []
    validation_labels = []

    for i, r in enumerate(reviews):
        cv = int(r["cv"])
        sent = sentiment_numerical_val[r["sentiment"]]
        content_string = ""
        for sentence in r["content"]:
            for word in sentence:
                content_string += word[0].lower() + separation

        if remove_punc:
            exclude = set(string.punctuation)
            content_string = ''.join(character for character in content_string if character not in exclude)

        if 0 < cv < split:
            training_set.append(content_string)
            training_labels.append(sent)
        else:
            validation_set.append(content_string)
            validation_labels.append(sent)

    return training_set, np.array(training_labels), validation_set, np.array(validation_labels)

def loadJSONData():
```





```
def train(self, batch_size, epochs, model, x_test, x_train, y_test, y_train):
    print("Train...")
    csv_logger = CSVLogger('cnn_lstm_log.csv', append=True, separator=";")
    history = model.fit(x_train, y_train, batch_size=batch_size, epochs=epochs, validation_data=(x_test, y_test), callbacks=[csv_logger])
    eval_epoch_history = model.evaluate(x_test, y_test, verbose=1)
    print('Loss:', eval_epoch_history[0])
    print('Accuracy:', "%0.2f%%" % (eval_epoch_history[1]*100))
    return history, eval_epoch_history
```

## Файл CNN.py

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.layers import Conv1D, MaxPooling1D
from tensorflow.keras.layers import Embedding
import time
import tensorflow.keras as K
from Model import Model

from MongoManager import saveToDB, loadModelFromDB, saveConfiguration

class CNN(Model):
    def __init__(self, *params):
        super().__init__(params)
        if len(params)==0: return
        self.max_words_number = params[0]
        self.max_review_len = params[1]

        self.configuration = params[2]
        self.embedding_size, \
        self.kernel_size, \
        self.filters, \
        self.pool_size, \
        self.dense_units1, \
        self.dense_units2, \
        self.batch_size, \
        self.epochs = self.configuration.getConfig()

    def toJSON(self):
        pass

    def defineModel(self, X_test, y_test, X_train, y_train, ):

        print()
        print(format("How to setup a CNN model for sentiment analysis in Keras", '^82'))

        print()
        print(X_train.shape)
        print(X_train)
        print()
        print(y_train.shape)
        print(y_train)
        print()
        print(X_test.shape)
        print(X_test)
        print()
        print(y_test.shape)
        print(y_test)

        model = Sequential()
        model.add(Embedding(self.max_words_number, self.embedding_size, input_length=self.max_review_len))
        model.add(Conv1D(filters=self.filters, kernel_size=self.kernel_size, padding='same', activation='relu'))
        model.add(MaxPooling1D(pool_size=self.pool_size))
        model.add(Flatten())
        model.add(Dense(self.dense_units1, activation='relu'))
        model.add(Dense(self.dense_units2, activation='sigmoid'))
        model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['acc'])
        model.summary()

        history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=self.epochs, batch_size=self.batch_size, verbose=1)

        eval_epoch_history = model.evaluate(X_test, y_test, verbose=1)
        print()

        print("Accuracy: %0.2f%%" % (eval_epoch_history[1] * 100))
        print()
        self.saveModel(model)
        return model, history, eval_epoch_history

    def runModel(self, model, userText, review_len):
        cnn_result, prediction, definedClass = self.doPrediction(model, userText, review_len)
        return cnn_result, prediction, definedClass

    def loadModel(self, filepath):
        model = loadModelFromDB(filepath)
        return model

    def saveModel(self, model, filename="cnn_model.h5"):
        fileID = saveToDB(filename, model)
        saveConfiguration("cnn_model", self.max_words_number, self.max_review_len, self.configuration.getConfig(), fileID)

    def doPrediction(self, model, userText, max_review_len):
        print("New review:" + userText)
        d = K.datasets.imdb.get_word_index()
        review = userText
        words = review.split()
        review = []
```

										Арк.
										75
Змн.	Арк.	№ докум.	Підпис	Дата	ДП 6303.00.000 ПЗ					

```

for word in words:
    if word not in d:
        review.append(2)
    else:
        review.append(d[word] + 3)

review = K.preprocessing.sequence.pad_sequences([review], truncating='pre', padding='pre',
maxlen=max_review_len)

prediction = model.predict(review)
print("Prediction (0 = negative, 1 = positive) = ", end="")
print("%0.4f" % prediction[0][0])
cnn_result = "Згорткова нейронна мережа"
definedClass=""
if prediction[0][0]>=0.5 : definedClass="Позитивний"
else:definedClass="Негативний"
return cnn_result,prediction[0][0],definedClass

```

### Файл AuthMongoManager.py

```

from bson import ObjectId
from pymongo import MongoClient
import hashlib

from UsersDataTable import makeUserTable

client = MongoClient("mongodb://localhost:27017/")
db = client['local']
userData = db['Users']

def saveUser(login,password):
    try:
        password = password.encode()
        print(password)
        passwordEncrypted = hashlib.md5(password).hexdigest()
        configToSave = {"Login": login, "Password": passwordEncrypted, "Access": True}
        userData.insert_one(configToSave)
    except:
        return "Error"
    return "Success"

def loadUser(inputLogin):
    login=""
    password=""
    access=""
    userList=list()
    for record in userData.find({"Login": inputLogin}):
        userList.append(record)
    if len(userList)!=0:
        login=userList[0]["Login"]
        password=userList[0]["Password"]
        access=userList[0]["Access"]
    return login,password,access

def loadAllUsers():
    tableResult=""
    userList = list()
    for record in userData.find():
        userList.append(record)
    if len(userList) == 0:
        return None
    else:
        tableResult=makeUserTable(userList)
    return tableResult

def findRecordById(id):
    x=list()
    for record in userData.find({"_id": ObjectId(id)}):
        x.append(record)
    return x

def updateUserAccess(id,access):
    updateQuery = {"_id": ObjectId(id)}
    newAccess = {"$set": {"Access": access}}
    userData.update_one(updateQuery, newAccess)

```

### Файл UsersDataTable.py

```

from flask_table import Table, Col, BoolCol, ButtonCol

class UsersTable(Table):
    classes = ['table', 'table-striped', 'table-condensed', 'table-hover']
    _id=Col("Номер користувача,№")
    Login=Col("Логін користувача")
    Password=Col("Зашифрований пароль користувача", show=False)
    Access=BoolCol("Доступ до моделей",yes_display="Так", no_display="Ні")
    Action=ButtonCol("Змінити доступ", "change", url_kwargs=dict(id='_id'))

def makeUserTable(userList):
    userTable=UsersTable(userList)
    return userTable

```

Змн.	Арк.	№ докум.	Підпис	Дата











```

<textarea name="inputText" cols="50" rows="3" class="form-control" style="resize:None">The movie was awesome. I love it</textarea>
<br>
<div class="btn-group btn-group-justified">
<div class="btn-group"><button type="submit" class="btn btn-primary">Почати навчання</button></div>
<div class="btn-group"><button type="submit" class="btn btn-primary" formaction="/load">Завантажити дані з бази даних</button>
</div>
</div>
</form>
</div>
</body>
</html>

```

## Файл DataTable.html

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
<body>
<div class="container">
<h2>Виберіть одну з моделей для проведення аналізу текстів на їх тональність </h2>
<h4>Моделі Рекурентної нейронної мережі з ДКЧП </h4>
<div style="max-height:250px; overflow-y: auto;">
{{ LSTM_table }}
</div>
<h4>Моделі Згорткової нейронної мережі </h4>
<div style="max-height:250px; overflow-y: auto;">
{{ CNN_table }}
</div>
<h4>Моделі Комбінованої нейронної мережі</h4>
<div style="max-height:250px; overflow-y: auto;">
{{ CNN_LSTM_table }}
</div>
<div align="center">
<form action="/run" method="POST">
<button type="submit" class="btn btn-primary btn-lg">Провести аналіз тексту</button>
</form>
</div>
</div>
</body>
</html>

```

## Файл login.html

```

<html>
<head>
<title>Flask Intro - login page</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
</head>
<body>
<div class="container">
<h1>Сторінка авторизації</h1>
<br>
<h4>Введіть, будь ласка, свій логін та пароль:</h4>
<form action="/login" method="POST">
<table class="table-condensed">
<tr>
<td><p>Логін:</p></td>
<td>
<input type="text" class="form-control" placeholder="Username" name="username" value="{{
request.form.username }}">
</td>
</tr>
<tr>
<td><p>Пароль:</p></td>
<td>
<input type="password" class="form-control" placeholder="Password" name="password" value="{{
request.form.password }}">
</td>
<td><input class="btn btn-default" type="submit" value="Ввійти"></td>
</tr>
</table>
<br>
<td colspan="2"><h5>Новий користувач? <a methods="GET" href="{{ url_for('register') }}">Натисніть сюди</a></h5></td>
</tr>
</table>
<br>
<p class="error"><strong>Помилка:</strong> {{ error }}
</p>
</form>
</div>
</body>
</html>

```

										Арк.
										82
Змн.	Арк.	№ докум.	Підпис	Дата						



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”  
Кафедра автоматизованих систем обробки інформації і управління

**УЗГОДЖЕНО**

**Керівник проєкту**

\_\_\_\_\_ Олена ЖДАНОВА

(підпис)

(вл. ім'я, прізвище)

“13” квітня 2020 р.

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

\_\_\_\_\_ Олександр ПАВЛОВ

(підпис)

(вл. ім'я, прізвище)

“14” квітня 2020 р.

Інформаційна технологія визначення тональності текстів

**ТЕХНІЧНЕ ЗАВДАННЯ**

Шифр *ДП 6303.01.1081-с ТЗ*

на 9 сторінках

Київ – 2020 року

# ЗМІСТ

<b>1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....</b>	<b>3</b>
<b>1.1 Повне найменування системи .....</b>	<b>3</b>
<b>1.2 Найменування організації-замовника та організацій-         учасників робіт.....</b>	<b>3</b>
<b>1.3 Перелік документів, на підставі яких створюється         система.....</b>	<b>3</b>
<b>1.4 Планові терміни початку і закінчення роботи зі створення         системи.....</b>	<b>3</b>
<b>2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ .....</b>	<b>4</b>
<b>2.1 Призначення системи.....</b>	<b>4</b>
<b>2.2 Цілі створення системи.....</b>	<b>4</b>
<b>3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ.....</b>	<b>6</b>
<b>4 ВИМОГИ ДО ПРОГРАМИ.....</b>	<b>8</b>
<b>4.1 Вимоги до функціональних характеристик.....</b>	<b>8</b>
<b>4.2 Вимоги до надійності.....</b>	<b>8</b>
<b>4.3 Вимоги до складу і параметрів технічних засобів.....</b>	<b>8</b>
<b>5 СТАДІЇ І ЕТАПИ РОЗРОБКИ.....</b>	<b>9</b>
<b>6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....</b>	<b>10</b>
<b>6.1 Об'єм та види випробувань.....</b>	<b>10</b>

					<b>ДП 6303.01.000 ТЗ</b>			
<i>Зм.</i>	<i>Арк.</i>	<i>Прізвище</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Блінков Є.М.</i>			<i>Інформаційна технологія визначення тональності текстів</i>	<i>Літ.</i>	<i>Лист</i>	<i>Листів</i>
<i>Перевірив.</i>		<i>Жданова О.Г.</i>					2	10
<i>Н. кон.</i>		<i>Проскура С.Л.</i>				<i>КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63</i>		
<i>Затв.</i>		<i>Павлов О.А.</i>						

## 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

### 1.1 Повне найменування системи

Інформаційна технологія визначення тональності текстів, далі –  
Інформаційна технологія.

### 1.2 Найменування організації-замовника та організацій-учасників робіт

Замовником системи є ТОВ «Інфопульс Україна». Розробником системи  
є Блінков Євген Миколайович.

### 1.3 Перелік документів, на підставі яких створюється система

Підставою для розробки інформаційної технології є завдання на  
переддипломну практику.

### 1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку робіт по створенню інформаційної технології  
– 13.04.2020.

Плановий робіт завершення робіт по створенню інформаційної  
технології – 01.06.2020.

					ДП 6303.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

## 2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ

### 2.1 Призначення системи

Призначенням даної інформаційної технології є автоматизація процесів проведення аналізу тональності текстів, застосовуючи різні алгоритми, та порівняння ефективності цих алгоритмів.

### 2.2 Цілі створення системи

Ціллю створення даної інформаційної технології є підвищення ефективності аналізу тональності тексту. Дану ціль можна досягти за умови, якщо програмний продукт буде виконувати наступні задачі:

- з) завантаження бази даних текстів для тренування моделей;
- и) очистка та підготовка даних;
- к) встановлення конфігурації моделей;
- л) запуск тренування моделей;
- м) здійснення аналізу тональності тексту;
- н) формування та відображення звітності;
- ж) порівняння ефективності різних моделей.

Для реалізації задач з пунктів г) та д) розробнику слід виконати наступні кроки:

- з) ознайомитися з існуючими підходами аналізу тональності текстів;
- и) розглянути приклади застосування даних підходів;
- к) обрати декілька підходів сентиментального аналізу, які здатні дати найточнішу оцінку емоційної складової тексту;
- л) проаналізувати можливості покращення обраних підходів;
- м) розробити систему з використанням обраних алгоритмів аналізу тональності тексту;

					ДП 6303.01.000 ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

- н) створити план тренування групи моделей аналізу тональності;
- о) створити план запуску кожної моделі з групи для проведення аналізу тексту.

					ДП 6303.01.000 ТЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Основними об'єктами автоматизації є процеси обробки вхідних даних та проведення аналізу тональності тексту, застосовуючи різні алгоритми, а також формування звітності з ефективності. Для успішного функціонування системи доступ до неї повинен бути розмежований між адміністратором та користувачем. Тому, слід виділити функції та обов'язки користувача та адміністратора в системі.

Адміністратор системи: надає користувачам доступ до моделей визначення тональності тексту або ж відкликає його.

Користувачі: отримують доступ до API розроблюваної системи та до веб-сайту системи, на якому вони можуть вибрати моделі визначення тональності тексту та здійснити запуск їх тренування.

Крім цього, користувач має змогу завантажити моделі та їх конфігурацію з бази даних.

Також, користувач може вручну задати параметри конфігурації певної моделі або ж скористатися запропонованим варіантом конфігурації.

Після того як тренування всіх моделей завершиться, користувач зможе переглянути детальний звіт щодо їх ефективності.

Користувач може задати власний текст для визначення його тональності і також побачити звіт з результатами роботи алгоритмів.

Функції Адміністратора:

- в) надання доступу користувачам до моделей;
- г) відкликання доступу у користувачів до моделей.

Функції Користувача:

- з) отримання доступу до API системи та до його веб-сайту;
- и) вибрати моделі визначення тональності тексту;
- к) запуск тренування моделей;

					ДП 6303.01.000 ТЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

- л) завантаження моделей та їх конфігурації з бази даних;
- м) встановлення параметрів конфігурації;
- н) задання власного тексту для його аналізу;
- о) перегляд звіту з результатами роботи алгоритмів та їх ефективності.

					ДП 6303.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

## 4 ВИМОГИ ДО ПРОГРАМИ

### 4.1 Вимоги до функціональних характеристик

Розроблений програмний продукт повинен виконувати такі функції:

- функція завантаження бази даних текстів для тренування моделей;
- функція очистки та підготовки даних;
- функція встановлення конфігурації моделей;
- функція запуску тренування моделей;
- функція здійснення аналізу тональності тексту;
- функція формування та відображення звітності;
- функція порівняння ефективності різних моделей.

### 4.2 Вимоги до надійності

Програмний продукт повинен коректним чином виправляти можливі збої та неполадки у системі, які можуть виникнути при підключенні до бази даних або при здійсненні аналізу тональності вхідного тексту. Крім цього, інформація щодо облікових записів адміністратора та користувачів повинна бути конфіденційною, щоб уникнути потенційних проблем з авторизацією в системі.

### 4.3 Вимоги до складу і параметрів технічних засобів

Склад технічних засобів визначається наявністю ЕОМ, яка має задовольняти наступні вимоги:

- наявність процесора з тактовою частотою не менше 1ГГц;
- наявність оперативної пам'яті місткістю не менше 2 ГБ;
- наявність жорсткого диску з вільним об'ємом пам'яті не менше 1 ГБ;
- встановлена операційна система версії Windows 7 або вище;
- наявність монітору та підключення до мережі Інтернет.

					ДП 6303.01.000 ТЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

## 5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

- а) Ознайомлення із постановкою завдання;
- б) збір та огляд спеціалізованої літератури;
- в) огляд існуючих реалізацій та підходів до аналізу тональності тексту;
- г) ознайомлення з обраними підходами;
- д) завантаження з мережі Інтернет бази даних для тренування моделей, яка знаходиться у публічному доступі;
- е) розробка програмного продукту;
- ж) тестування та налагодження програми;
- з) розробка експериментів для дослідження алгоритмів аналізу тональності тексту;
- и) формування звітності;
- к) здача готового програмного продукту.

					ДП 6303.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

## 6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Для контролю замовником розробленого програмного продукту всі випробування повинні бути проведеними на його ЕОМ. Крім цього, приймання готової роботи повинно відбутися не пізніше 17.05.2020.

### 6.1 Об'єм та види випробувань

Загалом, об'єм випробувань складається з двох основних частин:

- тестові випробування – призначаються для демонстрації замовнику як в цілому працює програмний продукт при певних заготовлених вхідних даних;
- експериментальні випробування – проводяться на ЕОМ замовника для відображення ефективності різних алгоритмів аналізу тональності за різних умов та різних вхідних даних.

					ДП 6303.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

***Звіт з результатами перевірки на співпадіння***

---

Київ – 2020 року

Власник документу:  
Попенко Володимир Дмитрович

Дата перевірки:  
04.06.2020 18:35:29 EEST

Дата звіту:  
04.06.2020 20:40:37 EEST

ID перевірки:  
1003783784

Тип перевірки:  
Doc vs Internet + Library

ID користувача:  
77149

Назва документу: Vlinkov\_bachelor

ID файлу: 1003798202 Кількість сторінок: 60 Кількість слів: 10344 Кількість символів: 76093 Розмір файлу: 1.64 MB

## 4.38% Схожість

Найбільша схожість: 2.36% з джерело бібліотеки, ID файлу: 1000018131

3.14% Схожість з Інтернет джерелами 74 Page 62

3.93% Текстові збіги по Бібліотеці акаунту 380 Page 63

## 0.21% Цитат

Цитати 1 Page 64

Вилучення переліку посилань викинено

## 0% Вилучень

Вилучений текст відсутній

## Підміна символів

Заміна символів 1

# **Графічний матеріал до дипломного проєкту**

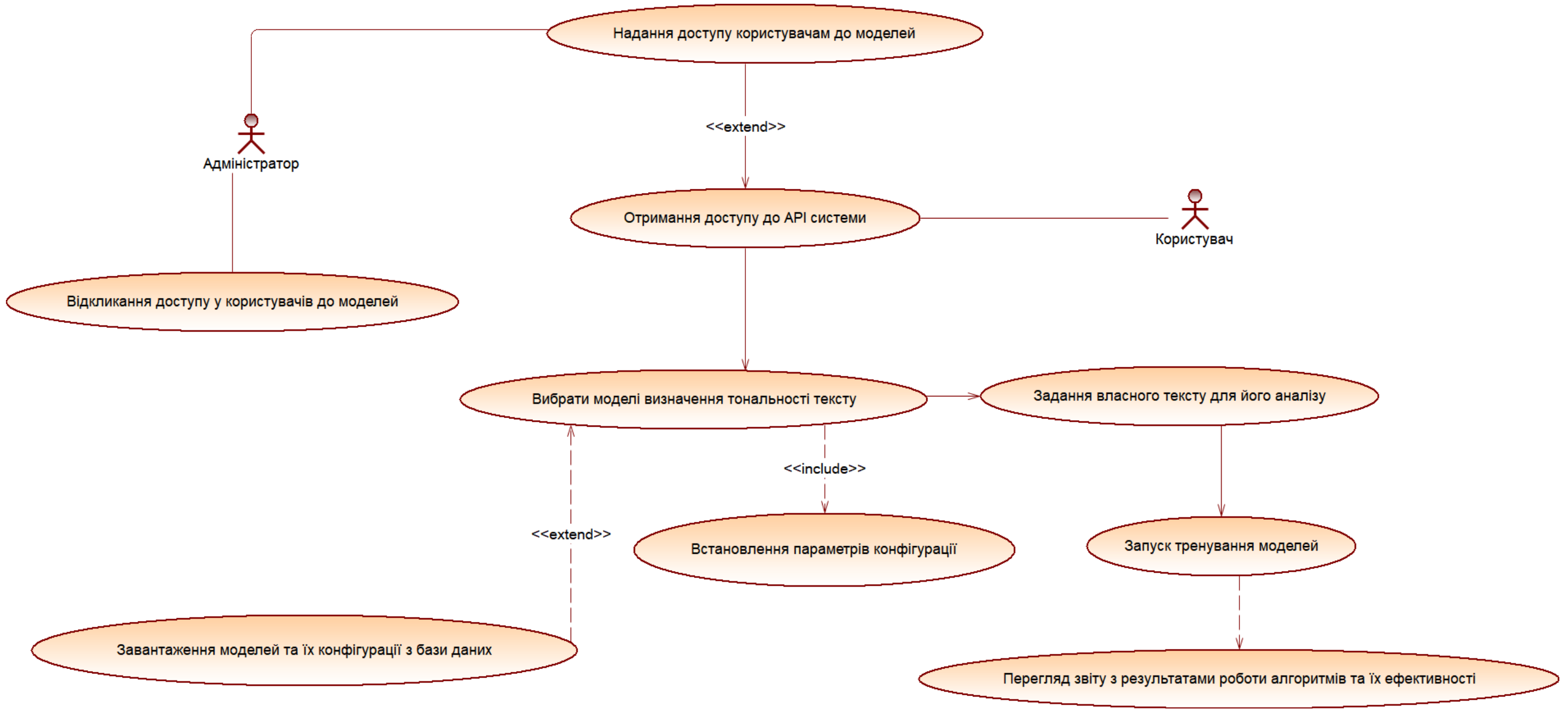
на тему: Інформаційна технологія визначення тональності текстів

---

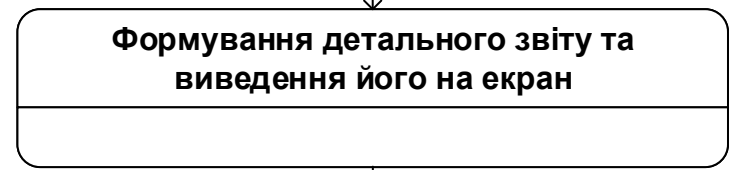
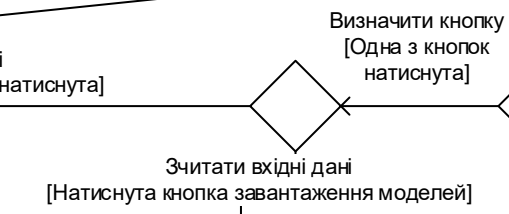
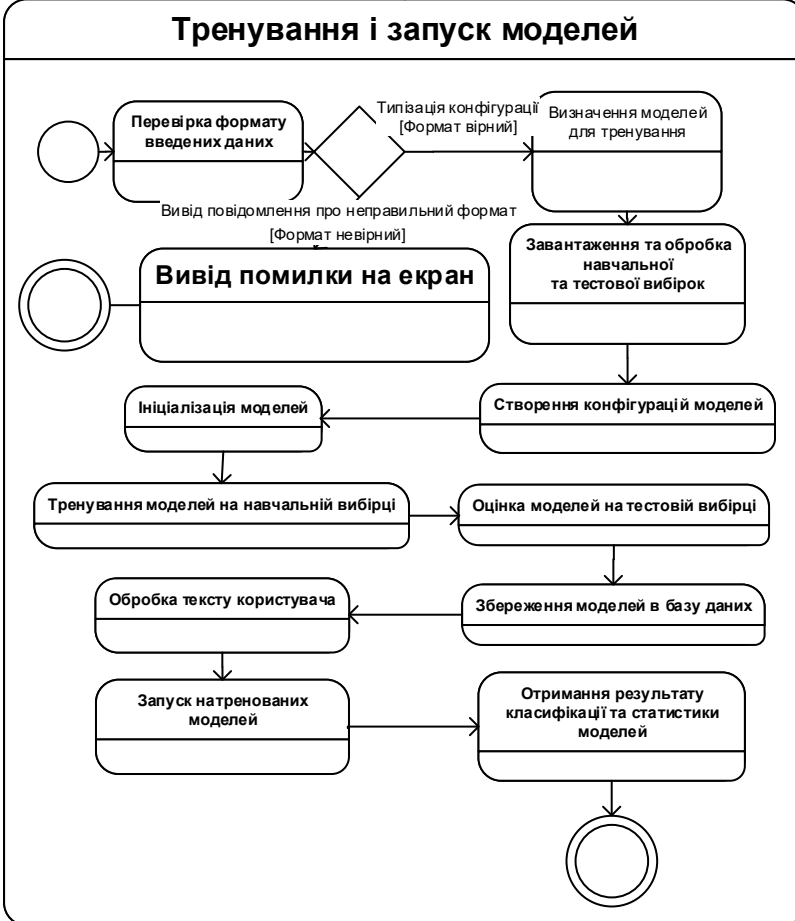
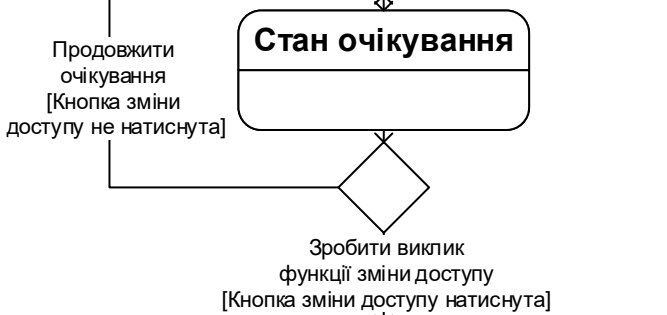
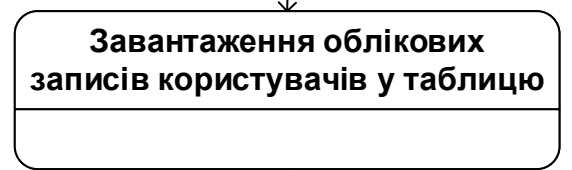
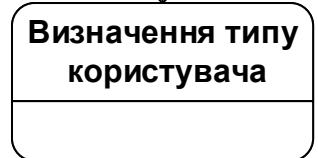
---

---

Київ – 2020 року



					ДП 6303.02.000 ССВ			
					<b>Схема структурна варіантів використання</b>	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Блінков Є.М.						
Перевірив		Жданова О.Г.						
Т. кон.						Аркуш 1	Аркушів 1	
Н. кон.		Проскура С.Л.			<i>Інформаційна технологія визначення тональності текстів</i>	<i>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63</i>		
Затвердив		Жданова О.Г.						



					ДП 6303.03.000 ССС		
Зм.	Арк.	№ документа	Підпис	Дата	Схема структурна станів		
Розробив	Блінков Є.М.						
Перевірив	Жданова О.Г.				Літера		Маса
Т. кон.					Аркуш 1		Масштаб
Н. кон.	Проскура С.Л.				Аркушів 1		
Затвердив	Жданова О.Г.				Інформаційна технологія визначення тональності текстів		
					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63		

Configuration	
<u>id</u>	<u>ObjectID</u> <pk>
ModelName	String
TopWords	Int32
MaxReviewLength	Int64
Configuration	Array
ModelFileName	ObjectID <fk>

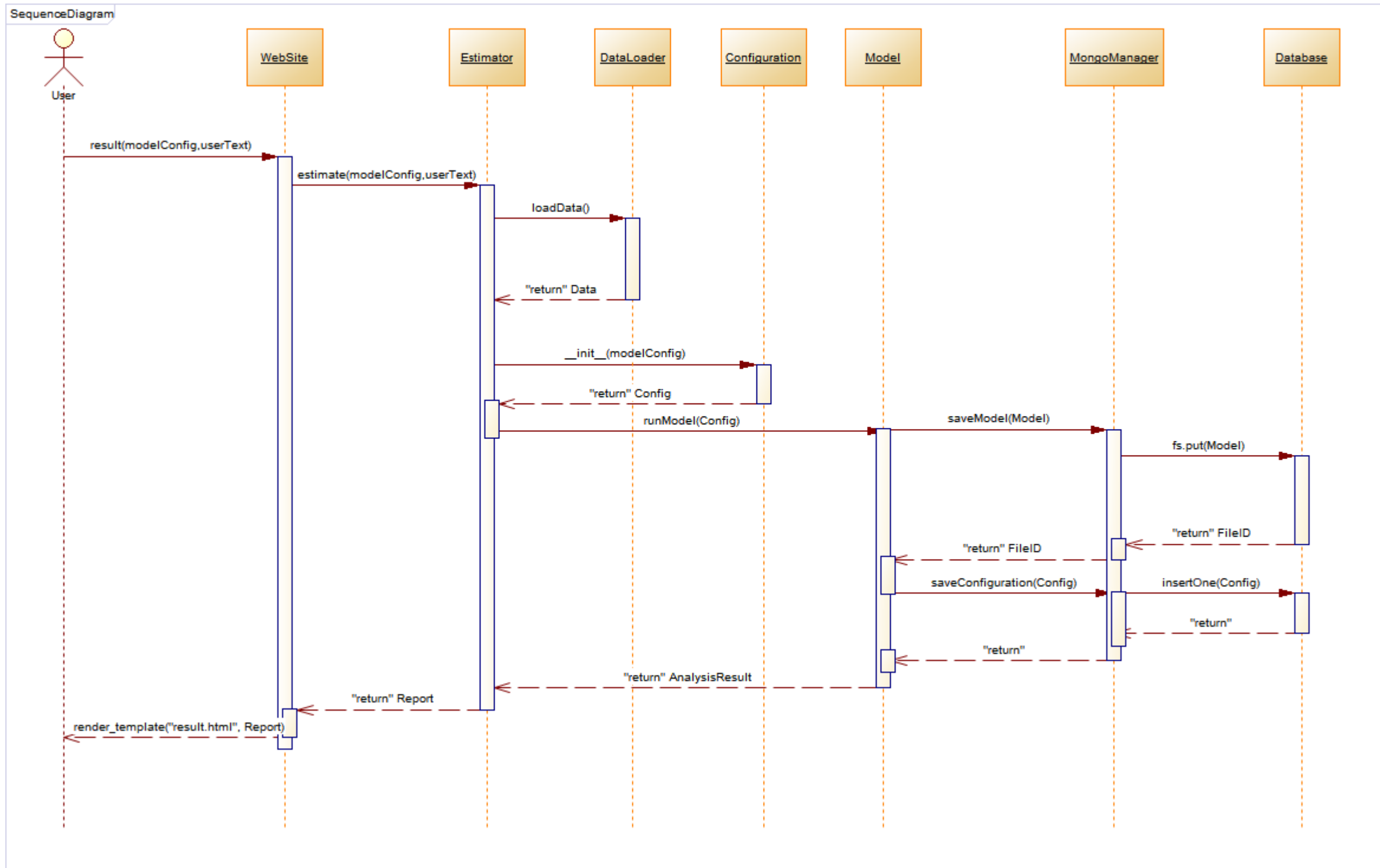
FK\_CONFIGUR\_REFERENCE\_FS.FILES

fs.files	
filename	String
length	Int64
Content_type	String
Upload_date	DATE
<u>md5</u>	<u>ObjectID</u> <pk>
metadata	String

Users	
<u>id</u>	<u>ObjectID</u> <pk>
Login	String
Password	String
Access	Bool

					ДП 6303.04.000 СБД			
					Схема бази даних	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Блінков Є.М.						
Перевірив		Жданова О.Г.						
Т. кон.						Аркуш 1	Аркушів 1	
Н. кон.		Проскура С.Л.			Інформаційна технологія визначення тональності текстів	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63		
Затвердив		Жданова О.Г.						





					ДП 6303.06.000 ССП			
Зм.	Арк.	№ документа	Підпис	Дата	<b>Схема структурна послідовності</b>	Літера	Маса	Масштаб
Розробив		Блінков Є.М.						
Перевірів		Жданова О.Г.						
Т. кон.						Аркуш 1	Аркушів 1	
Н. кон.		Проскура С.Л.				Інформаційна технологія визначення тональності текстів		
Затвердив		Жданова О.Г.			КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63			

# Рішення з математичного забезпечення

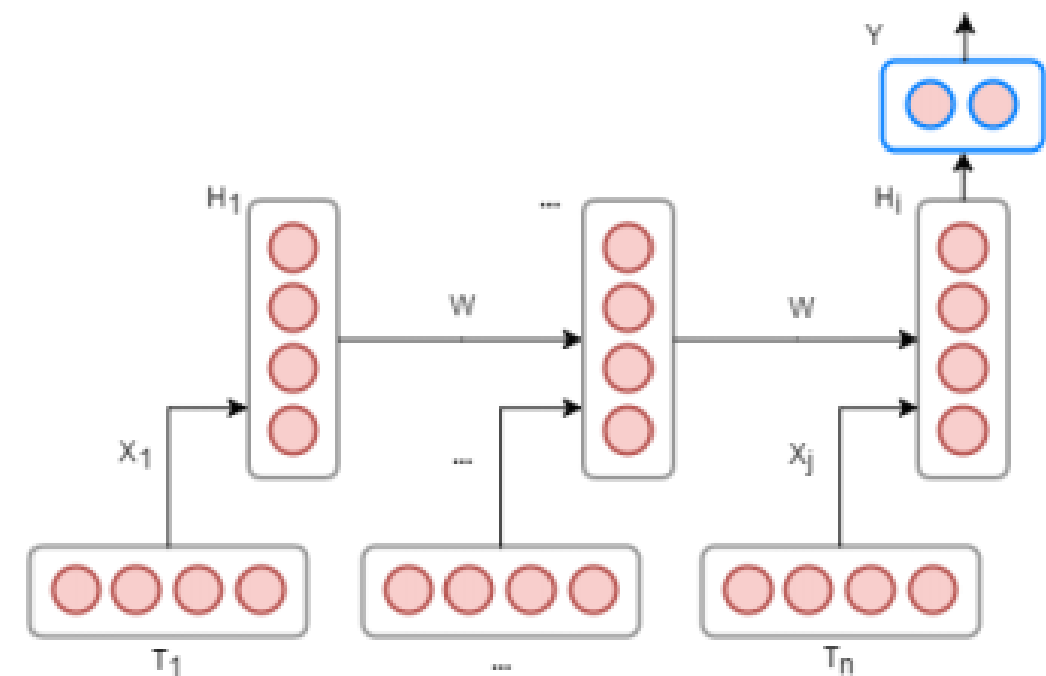
*Схема алгоритму роботи рекурентної нейронної мережі з технологією довгої короткочасної пам'яті (ДКЧП)*

**Крок 1:** За допомогою Word2Vec з кожного слова тексту утворити векторне представлення

**Крок 2:** Отримані слова  $\{T_1, \dots, T_n\}$  один за одним ввести у ДКЧП

**Крок 3:** В кожен момент часу  $j$  вихід  $W$  прихованого шару передати назад у прихований шар  $H_i$  разом з наступним вхідним вектором  $x_{j+1}$  у наступний момент часу  $j+1$ .

**Крок 4:** Останній вихід  $W$  передати вихідному шару нейронної мережі



Демонстраційний плакат до дипломного проекту

„Інформаційна технологія визначення тональності текстів”

Виконав студент гр. ІС-63

Блінков Є.М.

Керівник ДП

Жданова О.Г.

### Веб-сторінка з вибором моделей для їх навчання

Виберіть модель визначення тональності тексту для її навчання

Список моделей:	Конфігурація моделі:	Опис необхідних параметрів:
<input checked="" type="checkbox"/> Наївний Баєсів класифікатор		
<input type="checkbox"/> Нейронна мережа з ДКЧП	32;True;100;0.2;0.2;1;1	1. Довжина векторного представлення тексту, 2. Маска для нульових елементів векторного представлення тексту, 3. Кількість нейронів у шарі ДКЧП.
<input type="checkbox"/> Комбінована нейронна мережа (ДКЧП+ЗНМ)	32;3;32;2;100;32;5;0.0	1. Довжина вихідних векторів з шару Embedding, 2. Розмір вікна згортки, 3. Кількість фільтрів для аналізу векторів слів.
<input type="checkbox"/> Згорткова нейронна мережа (ЗНМ)	32;3;32;2;250;1;128;3	6. Кількість нейронів у другому звичайному шарі нейронної мережі, 7. Розмір вибірки, для здійснення однієї ітерації тренування моделі, 8. Кількість періодів навчання нейронної мережі

Кількість найпоширеніших слів у словнику:

Кількість слів у текстах:

Введіть текст для визначення його тональності:

### Веб-сторінка з вибором моделей завантажених з бази даних

Виберіть одну з моделей для проведення аналізу текстів на їх тональність

Моделі Рекурентної нейронної мережі з ДКЧП :

Назва моделі	Кількість найчастіших слів	Довжина тексту	Конфігурація моделі	Назва файлу з моделлю	Вибрати модель
lstm_model	5000	100	[32, True, 100, 0.2, 0.2, 1, 128, 3]	5ec90a460dd9443be84e9fe5	<input type="button" value="Вибрати модель"/>
lstm_model	5000	100	[32, True, 100, 0.2, 0.2, 1, 128, 3]	5eca83040732a5322f80c7b9	<input type="button" value="Вибрати модель"/>
lstm_model	7000	100	[32, True, 100, 0.2, 0.2, 1, 128, 3]	5eca89f059749cb5748a48cb	<input type="button" value="Вибрати модель"/>
lstm_model	5000	100	[32, True, 100, 0.2, 0.2, 1, 128, 3]	5ecabeb46a1beadc2431478f	<input type="button" value="Вибрати модель"/>
lstm_model	5000	100	[32, True, 100, 0.2, 0.2, 1, 128, 3]	5ecacadc8e8380e6adedeb3a	<input type="button" value="Вибрати модель"/>
lstm_model	5000	100	[32, True, 100, 0.2, 0.2, 1, 128, 3]	5ecac6c76f54de69f7469580	<input type="button" value="Вибрати модель"/>

Моделі Згорткової нейронної мережі :

Назва моделі	Кількість найчастіших слів	Довжина тексту	Конфігурація моделі	Назва файлу з моделлю	Вибрати модель
cnn_model	5000	100	[32, 3, 32, 2, 250, 1, 128, 3]	5ecac50462fbeb665bddbe71	<input type="button" value="Вибрати модель"/>
cnn_model	5000	100	[32, 3, 32, 2, 250, 1, 128, 3]	5ecac6152473d000afe899a7	<input type="button" value="Вибрати модель"/>
cnn_model	5000	100	[32, 3, 32, 2, 250, 1, 128, 3]	5ecac7c7038f0a34d089ec5a	<input type="button" value="Вибрати модель"/>
cnn_model	5000	100	[32, 3, 32, 2, 250, 1, 128, 3]	5ecac883fd20c4282947721f	<input type="button" value="Вибрати модель"/>
cnn_model	5000	100	[32, 3, 32, 2, 250, 1, 128, 3]	5ecaca39ae8380e6adedeb1d	<input type="button" value="Вибрати модель"/>
cnn_model	5000	100	[32, 3, 32, 2, 250, 1, 128, 3]	5ecace2e6f54de69f7469563	<input type="button" value="Вибрати модель"/>

Моделі Комбінованої нейронної мережі:

### Вигляд веб-сторінки з вибраними моделями

Виберіть одну з моделей для проведення аналізу текстів на їх тональність

Моделі Рекурентної нейронної мережі з ДКЧП :

Назва моделі	Кількість найчастіших слів	Довжина тексту	Конфігурація моделі	Назва файлу з моделлю
lstm_model	5000	100	[32, True, 100, 0.2, 0.2, 1, 128, 3]	5ec90a460dd9443be84e9fe5

Моделі Згорткової нейронної мережі :

Назва моделі	Кількість найчастіших слів	Довжина тексту	Конфігурація моделі	Назва файлу з моделлю
cnn_model	5000	100	[32, 3, 32, 2, 250, 1, 128, 3]	5ec909970dd9443be84e9fc8

Моделі Комбінованої нейронної мережі:

Назва моделі	Кількість найчастіших слів	Довжина тексту	Конфігурація моделі	Назва файлу з моделлю
cnn_lstm_model	5000	100	[32, 3, 32, 2, 100, 32, 5, 0.05, 1, 1]	5ec909760dd9443be84e9fbb

### Вигляд веб-сторінки авторизації

127.0.0.1:5000

#### Сторінка авторизації

Введіть, будь ласка, свій логін та пароль:

Логін:

Пароль:

Новий користувач? [Натисніть сюди](#)

### Вигляд веб-сторінки реєстрації

#### Реєстрація нового користувача

Введіть логін:

Введіть пароль:

Повторіть пароль:

### Вигляд веб-сторінки адміністратора системи

Користувачі системи:

Номер користувача, №	Логін користувача	Доступ до моделей	Змінити доступ
5ec842ec8d4e6cd2c158480d	Yevhen	Ні	<input type="button" value="Змінити доступ"/>
5ec90b0238518da602d7fe0a	Yura	Так	<input type="button" value="Змінити доступ"/>

					<b>ДП 6303.07.000 КЕ</b>			
Зм.	Арк.	№ документа	Підпис	Дата	<b>Креслення вигляду екранних форм</b>	Літера	Маса	Масштаб
Розробив		Блінков Є.М.						
Перевірів		Жданова О.Г.						
Т. кон.						Аркуш 1	Аркушів 2	
Н. кон.		Проскура С.Л.			<b>Інформаційна технологія визначення тональності текстів</b>	<b>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63</b>		
Затвердив		Жданова О.Г.						

Таблиця з результатами роботи алгоритмів

Результати роботи моделей визначення тональності тексту:

Введений користувачем текст: The movie was awesome. I love it

Назва моделі	Значення тональності	Визначений клас тексту
Рекурентна нейронна мережа з довгою короткочасною пам'яттю (ДКЧП)	0.699	Позитивний
Згортова нейронна мережа	0.487	Негативний
Комбінована нейронна мережа(ДКЧП+ЗНМ)	0.822	Позитивний

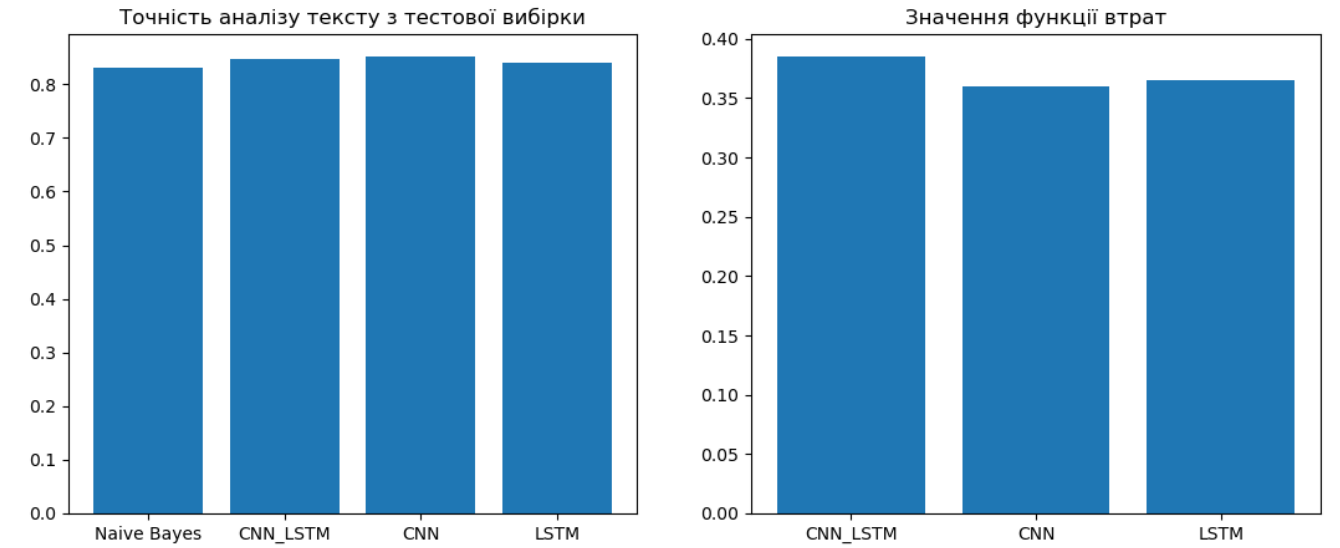
Фрагмент таблиці зі статистикою навчання та тестування моделей

Згортова нейронна мережа	Назва етапу	Статистика моделі			21.682902812957764
	Навчання	Номер епохи	Точність моделі	Значення функції втрат	
		0	0.73516	0.4842801664113998	
		1	0.89108	0.26686837924003604	
		2	0.92556	0.19378582264900207	
	Тестування	Номер епохи	Точність моделі	Значення функції втрат	
		0	0.85092	0.3598198759508133	

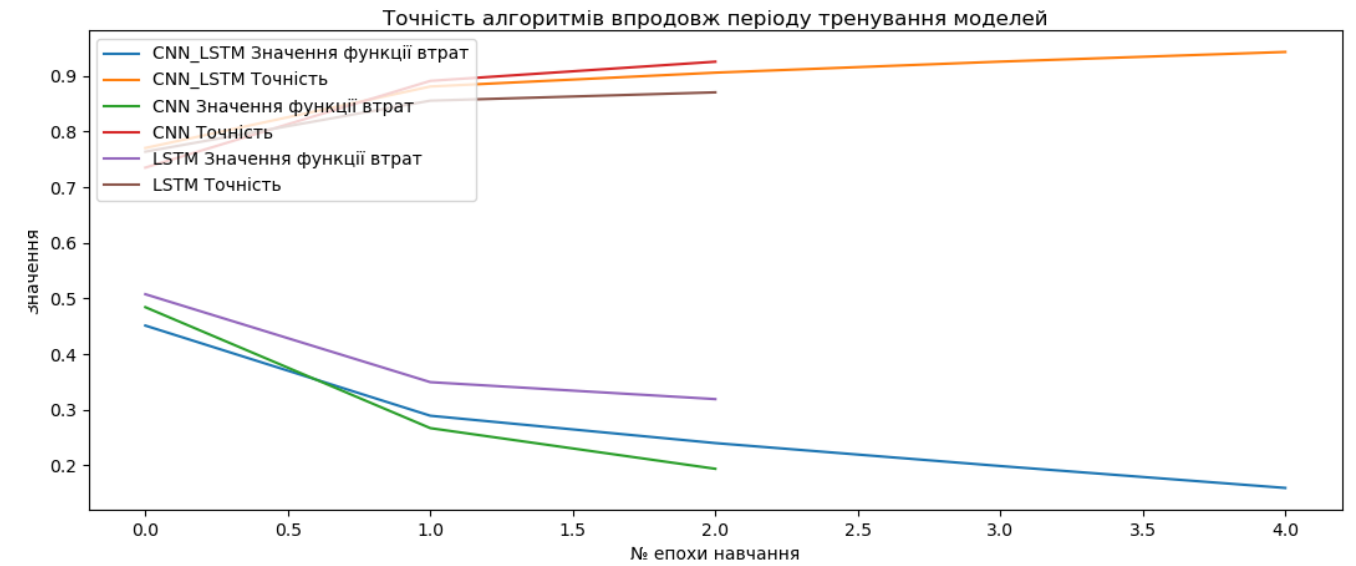
  

Рекурентна нейронна мережа з ДКЧП	Назва етапу	Статистика моделі			138.97327184677124
	Навчання	Номер епохи	Точність моделі	Значення функції втрат	
		0	0.7638	0.5075968775939942	
		1	0.85552	0.34953057083129885	
		2	0.87048	0.3190530140590668	
	Тестування	Номер епохи	Точність моделі	Значення функції втрат	
		0	0.84116	0.3648605983018875	

Гістограми точності класифікації текстів з тестової вибірки



Порівняльний графік збільшення точності та зниження значення функції втрат моделей нейронних мереж впродовж тренування



					ДП 6303.07.000 КЕ			
Зм.	Арк.	№ документа	Підпис	Дата	<b>Креслення вигляду екранних форм</b>	Літера	Маса	Масштаб
Розробив		Блінков Є.М.						
Перевірів		Жданова О.Г.						
Т. кон.						Аркуш 2	Аркушів 2	
Н. кон.		Проскура С.Л.			<b>Інформаційна технологія визначення тональності текстів</b>	<b>КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63</b>		
Затвердив		Жданова О.Г.						