МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ПАВЛЮК ВЛАДИСЛАВ ВІТАЛІЙОВИЧ

Допускається до захисту:
В. о. завідувача кафедри
інформаційних технологій,
к. т. н, доцент
_____О. В. Зелінська
«____»_____ 2024 р.

РОЗРОБКА САЙТУ НОВИН НА ПІДСТАВІ МАШИННОГО НАВЧАННЯ

Спеціальність 122 «Комп’ютерні науки»

Кваліфікаційна (магістерська) робота

Науковий керівник:
Н.Р. Веселовська, професор
кафедри інформаційних
технологій, д.т.н., професор
_____

Оцінка: ___/___/_____
(бали за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця 2024

АНОТАЦІЯ

Павлюк В.В. Дослідження сайту новин за допомогою методів машинного навчання.

Спеціальність 122 «Комп'ютерні науки», освітня програма «Розробка програмного забезпечення та комп'ютерна графіка», Донецький національний університет імені Василя Стуса, Вінниця, 2024.

У кваліфікаційній магістерській роботі розроблено новинний веб-сайт із застосуванням алгоритмів машинного навчання для класифікації, аналізу тональності новин, виявлення фейкових новин та персоналізації рекомендацій для користувачів. Основна увага приділена розробці моделей, заснованих на методах глибокого навчання, таких як BERT, а також створенню функціонального інтерфейсу для роботи з новинами. Результати дослідження підтверджують ефективність розробленої системи, забезпечуючи високу точність класифікації та покращуючи користувацький досвід.

Ключові слова: веб-сайт новин, машинне навчання, аналіз, взаємодія користувачів, аналіз настроїв, категоризація тем, аналіз тенденцій, Python, Django.

56 с., 3 табл., 20 рис., 2 дод., 31 джерел.

ANNOTATION

Pavliuk V.V. Research of a News Website Using Machine Learning Methods. Specialty 122 "Computer Science," educational program "Software Development and Computer Graphics," Donetsk National University named after Vasyl Stus, Vinnytsia, 2024.

The master's qualification thesis presents the development of a news website utilizing machine learning algorithms for classification, sentiment analysis, fake news detection, and personalized recommendations for users. The focus is on creating models based on deep learning methods such as BERT, as well as designing a functional interface for interacting with news content. The research results confirm the efficiency of the developed system, ensuring high classification accuracy and enhancing the user experience.

Keywords: news website, machine learning, analysis, user interaction, sentiment analysis, topic categorization, trend analysis, Python, Django.

56 pages, 3 tables, 20 figures, 2 appendices, 31 references.

ЗМІСТ

ВСТУП

Актуальність теми дослідження. Дослідження новинних веб-сайтів із використанням методів машинного навчання набуває особливої актуальності в наш час, коли обсяги інформації постійно зростають, а її обробка потребує високої ефективності. Новинні веб-сайти є важливим джерелом щоденного інтернет-трафіку, створюючи значну кількість контенту, що потребує грамотного управління, залучення аудиторії та оптимальної подачі.

Методи машинного навчання мають потенціал значно покращити ці процеси. Зокрема, автоматична класифікація новин за категоріями дозволяє пропонувати користувачам персоналізований і більш релевантний контент, що підвищує їх залученість. Аналіз тональності текстів допомагає виявити громадську думку на різні теми та оптимізувати створення новинного контенту відповідно до інтересів аудиторії.

Зростання обсягів фейкових новин та дезінформації створює додатковий виклик, який можна вирішити за допомогою алгоритмів машинного навчання для автоматичного виявлення й фільтрації недостовірного контенту. Це сприяє зміцненню довіри до новинних ресурсів та підвищує їхню репутацію.

Це дослідження також робить внесок у ширшу сферу машинного навчання та обробки природної мови, застосовуючи ці методи до даних реального світу в практичних умовах. Висновки та методології, розроблені в ході цього дослідження, також можуть бути застосовані до інших областей, які мають справу з великими обсягами текстових даних, таких як платформи соціальних мереж, блоги та онлайн-форуми.

Розробка сайту новин із застосуванням алгоритмів машинного навчання дозволить автоматизувати обробку даних, забезпечити персоналізацію новин для користувачів і підвищити точність класифікації новинного контенту. Дослідження також спрямоване на виявлення фейкових новин, що є особливо важливим у сучасному інформаційному просторі.

Об’єкт дослідження. Об'єктом дослідження у даній дипломній роботі сайт новин. Зокрема, це Процеси аналізу та автоматизації обробки новинного контенту за допомогою машинного навчання.

Предмет дослідження. Предметом дослідження в цій дипломній роботі є застосування методів машинного навчання для аналізу та розуміння даних, отриманих із сайту новин.

Це включає:

Алгоритми та методи машинного навчання, що використовуються для класифікації, аналізу тональності, виявлення фейкових новин і формування рекомендацій на новинних сайтах..

Моделі машинного навчання для прогнозування поведінки користувачів, наприклад взаємодії зі статтями (наприклад, оцінки «подобається», поширення, коментарі), на основі різних факторів, як-от вміст статті, час публікації тощо.

Методи візуалізації даних для ефективного представлення результатів аналізу та надання розуміння, яке можна інтерпретувати.

Розробка та впровадження системи для автоматизації цих процесів для аналізу нових даних у режимі реального або майже в режимі реального часу, коли вони стають доступними на сайті новин.

Загальна мета полягає в тому, щоб використовувати машинне навчання, щоб покращити наше розуміння даних новинного сайту та отримати корисну інформацію.

Мета. Розробка новинного веб-сайту із застосуванням алгоритмів машинного навчання для автоматизації обробки новинного контенту, виявлення фейкових новин, аналізу тональності та створення рекомендаційної системи.

Завдання дослідження. Для досягнення поставленої мети були вирішені наступні завдання:

1. Вивчити існуючі підходи до автоматизації обробки новинного контенту.
2. Моделювання тем: зібрані статті новин з різних веб-сайтів новин і попередня обробка реалізовані методи машинного навчання, щоб автоматично класифікувати статті новин за різними темами. Це допомагає краще організувати та персоналізувати вміст.
3. Створити функціонал персоналізованих рекомендацій для користувачів.
4. Оцінка та аналіз: проведено комплексну оцінку реалізованих моделей машинного навчання з точки зору їх точності, ефективності та впливу на залучення користувачів.
5. Презентація результатів: дослідження були систематично організовані та представлені, висвітлюючи ключові ідеї та покращення, досягнуті у функціональності новинних веб-сайтів за допомогою методів машинного навчання.

Виконуючи ці завдання, дослідження має на меті продемонструвати потенціал методів машинного навчання для покращення роботи та залучення користувачів новинних веб-сайтів.

Наукова новизна дослідження. Вперше запропоновано інтеграцію моделі BERT із системами класифікації новинного контенту для підвищення точності аналізу та забезпечення виявлення недостовірної інформації в реальному часі. Розроблено адаптивну систему рекомендацій, яка враховує індивідуальні інтереси користувачів.

Практична цінність. Результати дослідження можуть бути застосовані у новинних агрегаторах для автоматизації обробки контенту, покращення персоналізації та підвищення довіри до новинних ресурсів.

Методи дослідження. Для досягнення мети використано методи машинного навчання, зокрема глибокого навчання (модель BERT), аналізу тональності та класифікації текстів. Для реалізації системи застосовано фреймворк Django та мову Python.

Робота складається зі вступу, 3 розділів, висновку, списку літературних джерел, який складається 31 елементу. Робота містить 20 рисунків. Загальний обсяг роботи становить 65 сторінок.

РОЗДІЛ 1. ОГЛЯД СТАНУ ДОСЛІДЖЕНЬ НОВИННИХ САЙТІВ

1.1 Використані підходи

Метою даного дослідження є розробка та впровадження методів аналізу новинних сайтів на підставі машинного навчання. Це включає в себе аналіз існуючих методів, їх адаптацію та вдосконалення для специфічних потреб аналізу новинного контенту.

Дане дослідження має значущість для розвитку галузі аналізу даних та інформаційних технологій. Розроблені методи можуть бути використані для покращення якості новинного контенту, підвищення точності класифікації та фільтрації новин, а також для запобігання розповсюдженню фейкової інформації. [13]. Приклад популярного сайту новин зображено на рисунку 1.1.

Рис. 1.1 – Сайт The New York Times [1]

Таким чином, дослідження методів аналізу сайтів новин на підставі машинного навчання є важливим та актуальним завданням, яке має значний потенціал для розвитку сучасних інформаційних технологій. Враховуючи швидкий темп

розвитку медіа-середовища та постійне збільшення обсягів новинного контенту, ефективні методи автоматизації та аналізу стають необхідністю для забезпечення достовірності, актуальності та зручності інформації для користувачів. Крім того, удосконалення існуючих алгоритмів і розробка нових підходів дозволить не лише підвищити точність і швидкість аналізу, але й забезпечити захист від фейкових новин і дезінформації, що є критично важливим у сучасному інформаційному просторі. Використання машинного навчання в цій сфері відкриває нові можливості для персоналізації контенту, виявлення трендів та аналітики даних, що в кінцевому підсумку сприятиме підвищенню якості та ефективності надання новинних послуг.

Основні методи аналізу новинного контенту можна поділити на кілька категорій:

- Текстова аналітика та обробка природної мови (NLP): Використання алгоритмів обробки природної мови для аналізу текстового контенту новин, включаючи токенізацію, лематизацію, виділення сутностей та аналіз тональності. Серед основних інструментів можна відзначити бібліотеки NLTK, SpaCy, та інші.
- Класифікація та категоризація: Алгоритми машинного навчання, такі як наївний байєсівський класифікатор, метод опорних векторів (SVM), випадковий ліс (Random Forest), використовуються для класифікації новин за різними категоріями (наприклад, політика, спорт, економіка).
- Аналіз тональності: Методи аналізу тональності дозволяють визначити емоційний зміст новинних статей (позитивний, негативний або нейтральний). Вони широко використовуються для моніторингу громадської думки та аналізу соціальних медіа.
- Виявлення фейкових новин: Використання алгоритмів машинного навчання для виявлення недостовірних або фейкових новин. Це включає аналіз змісту статей, а також метаданих, таких як джерело новин, час публікації, авторство.

На сьогодні існує багато досліджень та інструментів, спрямованих на аналіз новинного контенту. Деякі з них включають:

- Дослідження класифікації новин: Вчені досліджують ефективність різних алгоритмів машинного навчання для класифікації новин. Наприклад, дослідження показують, що глибокі нейронні мережі, такі як LSTM і CNN, демонструють високу точність у завданнях класифікації тексту.
- Інструменти обробки природної мови: Популярні бібліотеки, такі як NLTK, SpaCy, та інші, забезпечують широкий спектр інструментів для обробки текстових даних, що значно спрощує аналіз новинного контенту.

- Виявлення фейкових новин: Дослідження в цій галузі активно розвиваються. Наприклад, моделі на основі глибоких нейронних мереж, такі як BERT, використовуються для аналізу тексту та виявлення фейкових новин з високою точністю.

Незважаючи на значний прогрес, існує низка проблем та викликів, які потребують подальшого дослідження:

- Якість даних: Недостатня якість даних може призвести до низької точності моделей машинного навчання. Необхідно розробляти методи очищення та попередньої обробки даних.
- Великі обсяги даних: Обробка великих обсягів новинного контенту вимагає значних обчислювальних ресурсів та ефективних алгоритмів.
- Мовні бар'єри: Аналіз новин на різних мовах потребує врахування специфіки кожної мови, що може ускладнювати процес обробки природної мови.

Етичні аспекти: Використання алгоритмів для аналізу новин викликає питання щодо етики, конфіденційності та можливих упереджень моделей

1.2 Завдання яке вирішує машинне навчання

Машинне навчання є однією з ключових технологій сучасного аналізу даних, що дозволяє автоматизувати обробку великих обсягів інформації, виявляти закономірності та робити прогнози на основі наявних даних. У контексті аналізу сайтів новин машинне навчання вирішує ряд специфічних завдань, які сприяють підвищенню точності та ефективності роботи з новинним контентом. [14]. Завдання машинного навчання у контексті аналізу сайтів новин:

- Класифікація новин: Одним із основних завдань є класифікація новинних статей за темами, такими як політика, економіка, спорт, технології тощо. Алгоритми машинного навчання, такі як метод опорних векторів (SVM), наївний байєсівський класифікатор та глибокі

нейронні мережі, використовуються для автоматичного визначення категорії кожної новини на основі її змісту.

- Аналіз тональності: Це завдання включає виявлення емоційного забарвлення тексту новинних статей, тобто визначення, чи є новина позитивною, негативною або нейтральною. Для цього використовуються алгоритми обробки природної мови (NLP), такі як рекурентні нейронні мережі (RNN) та BERT.
- Виявлення фейкових новин: В умовах поширення недостовірної інформації особливо актуальним є завдання виявлення фейкових новин. Машинне навчання дозволяє аналізувати текст, метадані (джерело, авторство) та інші фактори, щоб визначити ймовірність того, що новина є фейковою. Для цього використовуються моделі глибокого навчання та інші методи, зокрема логістична регресія та градієнтний бустинг.
- Рекомендаційні системи: Машинне навчання також використовується для розробки рекомендаційних систем, які пропонують користувачам новини на основі їхніх уподобань та історії переглядів. Алгоритми колаборативної фільтрації та контентно-орієнтовані методи допомагають забезпечити персоналізований підхід до подачі новин.
- Аналіз заголовків та змісту: Алгоритми машинного навчання аналізують заголовки та зміст новинних статей для виявлення ключових тем, трендів та інших важливих аспектів. Це завдання включає виділення сутностей, виявлення ключових слів та фраз, а також аналіз структури тексту.
- Моніторинг та прогнозування: Завдяки машинному навчанню можна здійснювати моніторинг новинних потоків у режимі реального часу та прогнозувати розвиток подій. Це корисно для виявлення потенційно важливих новин на ранніх стадіях їх розповсюдження.

Приклади використання:

- Google News: Використовує алгоритми машинного навчання для класифікації новин, аналізу тональності та персоналізації новинних стрічок для користувачів. (рис. 1.2).
- Facebook: Застосовує машинне навчання для виявлення фейкових новин та пропонування користувачам релевантного контенту на основі їхньої взаємодії з платформою (рис. 1.3).
- Reuters: Використовує алгоритми обробки природної мови для аналізу новинного контенту та виявлення трендів і ключових тем (рис. 1.4).

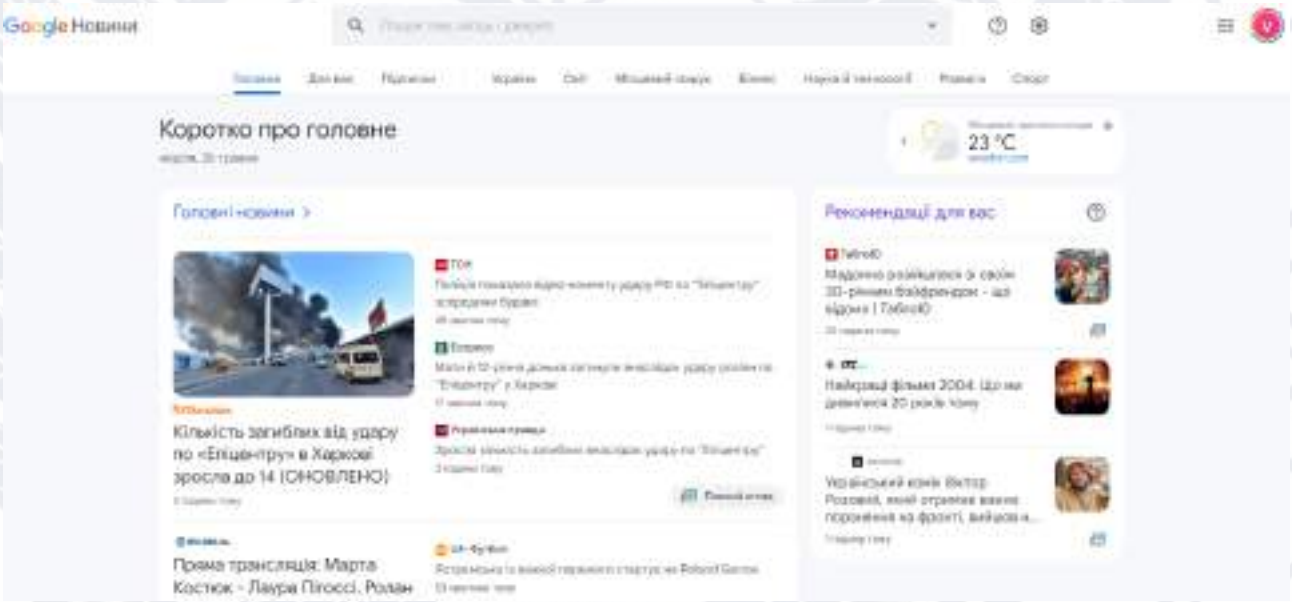

Рис. 1.2 Google News [15]

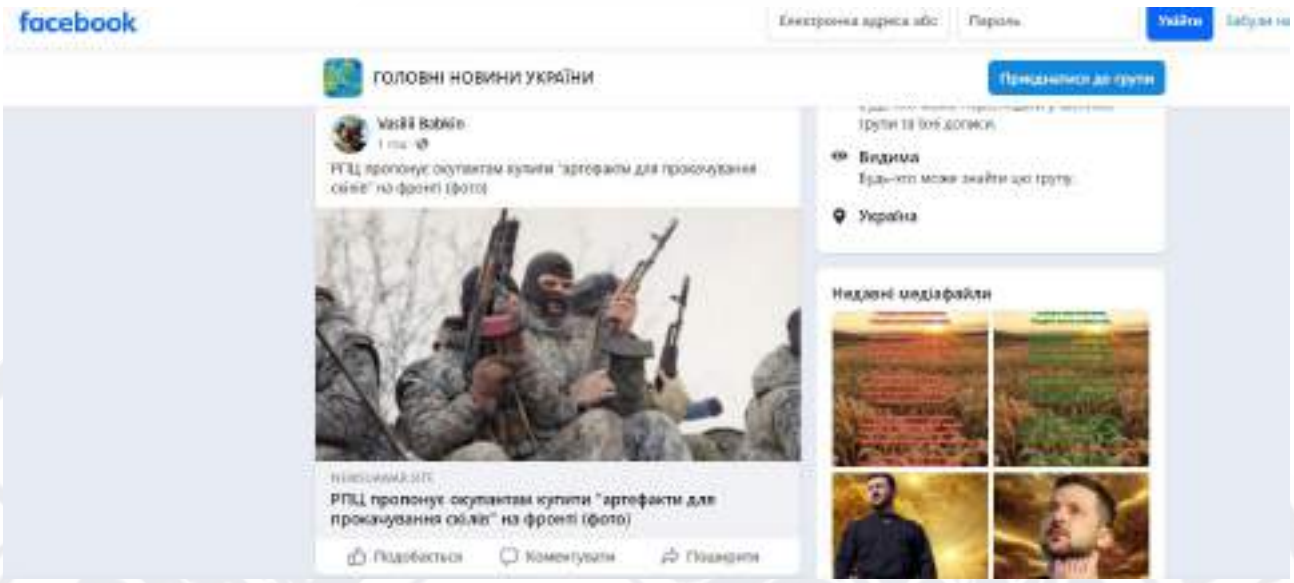

Рис. 1.3 Facebook

Рис. 1.4 Reuters

Машинне навчання вирішує широкий спектр завдань, пов'язаних з аналізом новинного контенту, від класифікації та аналізу тональності до виявлення фейкових новин та прогнозування подій. Використання цих технологій дозволяє автоматизувати обробку великих обсягів даних, підвищити точність та швидкість аналізу, а також забезпечити персоналізований підхід до подачі новин. Таким чином, машинне навчання є невід'ємною складовою сучасного аналізу новинних сайтів. [16]

1.3 Детальний огляд новинних сайтів, що використовують машинне навчання: Reuters, Google News.

Reuters є одним із провідних світових новинних агентств, яке широко використовує технології машинного навчання для аналізу, класифікації та подачі новин. В цьому розділі розглянемо детальніше, як Reuters інтегрує машинне навчання у свої процеси та які результати це приносить. Reuters використовує алгоритми машинного навчання для автоматичної класифікації новин за темами. [17] Це дозволяє користувачам швидко знаходити новини, що їх цікавлять, у різних категоріях, таких як політика, економіка, спорт, технології тощо. Для цього використовуються методи, як-от:

- Метод опорних векторів (SVM): Використовується для поділу новинних статей на різні категорії на основі текстового змісту.
- Глибокі нейронні мережі (Deep Learning): Застосовуються для обробки великих обсягів даних і складних текстових структур, що покращує точність класифікації.

Reuters використовує алгоритми обробки природної мови (NLP) для аналізу тональності новинних статей. Це дозволяє визначити емоційне забарвлення тексту, що є корисним для оцінки громадської думки та виявлення трендів. Наприклад:

- BERT (Bidirectional Encoder Representations from Transformers): Використовується для контекстного аналізу тексту, що дозволяє точно визначати тональність новинних статей.

Reuters активно працює над виявленням фейкових новин за допомогою машинного навчання. Це включає аналіз тексту, метаданих і інших характеристик новинних статей. Використовуються такі методи:

- Логістична регресія: Використовується для оцінки ймовірності того, що новина є фейковою на основі визначених характеристик.

– Градієнтний бустинг: Ефективний метод для виявлення складних закономірностей у даних, що дозволяє покращити точність виявлення фейкових новин.

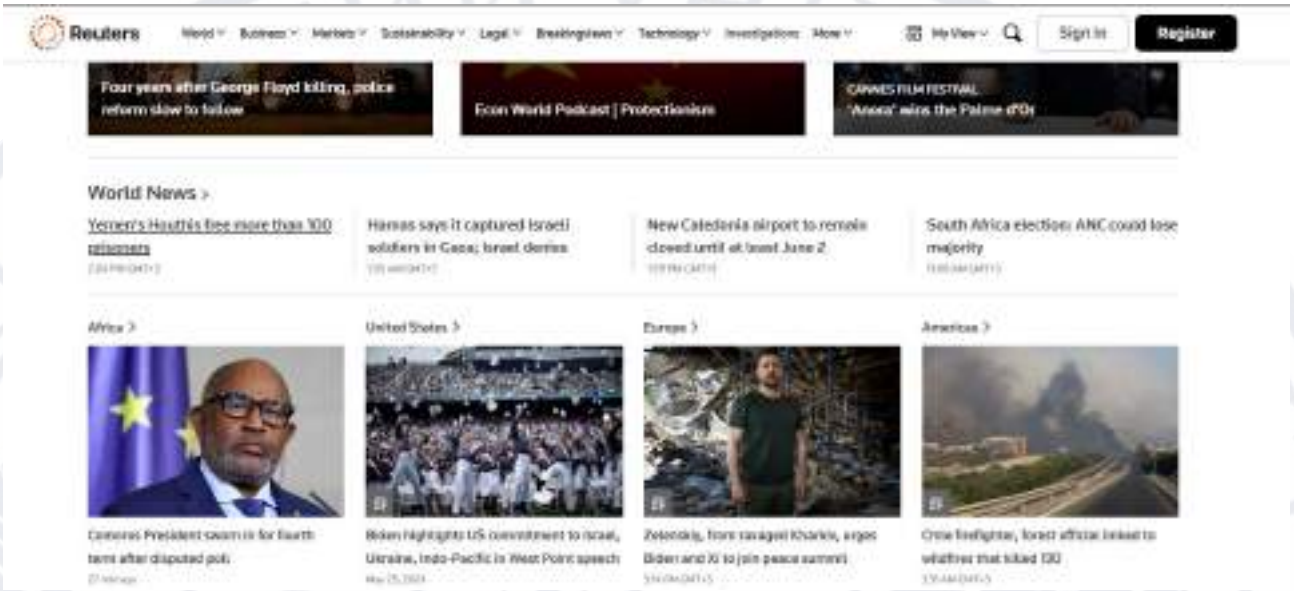

Рис. 1.5 – Головна сторінка сайту інформаційної системи Reuters [5]

Reuters використовує рекомендаційні системи для персоналізації подачі новин. Алгоритми машинного навчання аналізують історію переглядів користувачів і їх уподобання для пропонування релевантного контенту. Застосовуються такі методи:

– Колаборативна фільтрація: Метод, що базується на аналізі уподобань користувачів із подібними інтересами.
– Контентно-орієнтована фільтрація: Метод, що використовує інформацію про самі новини для формування рекомендацій.

Reuters також використовує машинне навчання для аналізу заголовків та змісту новинних статей, що дозволяє виявляти ключові теми та тренди. Це включає::

– Виділення сутностей (Named Entity Recognition, NER): Алгоритми виділяють і класифікують важливі сутності в тексті, такі як імена людей, організації, місця тощо.

- Аналіз ключових слів та фраз: Використання алгоритмів для виділення найважливіших слів і фраз у тексті, що допомагає зрозуміти основний зміст новини.

Впровадження машинного навчання дозволило Reuters значно підвищити ефективність та точність аналізу новинного контенту. Основні переваги включають:

- Швидкість обробки: Алгоритми машинного навчання дозволяють швидко обробляти великі обсяги новинного контенту, що забезпечує актуальність інформації.
- Точність: Використання сучасних методів машинного навчання, таких як глибокі нейронні мережі та BERT, дозволяє досягти високої точності в класифікації та аналізі новин.
- Персоналізація: Рекомендаційні системи забезпечують користувачам доступ до релевантного контенту, що підвищує їх задоволення від використання сайту

Reuters демонструє успішне впровадження технологій машинного навчання у свої процеси, що дозволяє підвищити якість та ефективність роботи з новинним контентом. Це приклад того, як машинне навчання може бути використане для розв'язання складних завдань у галузі медіа та інформаційних технологій.

Цей детальний огляд ілюструє, як новинний сайт може використовувати машинне навчання для покращення своїх сервісів і задоволення потреб користувачів.

Google News є одним із найпопулярніших новинних агрегаторів у світі, який використовує передові технології машинного навчання для збору, аналізу та подачі новин. У цьому розділі розглянемо, як Google News інтегрує машинне навчання у свої процеси, і які результати це приносить. [19]

Google News використовує алгоритми машинного навчання для автоматичного збору новин з різних джерел по всьому світу. Це дозволяє швидко і ефективно агрегувати новини з тисяч різних сайтів і забезпечувати користувачам доступ до найсвіжішої інформації.

Google News класифікує новини за різними категоріями, такими як світові новини, бізнес, технології, спорт та інші. Для цього використовуються такі методи машинного навчання:

- Кластеризація: Використовується для групування схожих новин разом, що полегшує користувачам пошук релевантної інформації.
- Глибокі нейронні мережі: Використовуються для аналізу тексту новин і автоматичної класифікації їх за темами.

Google News застосовує алгоритми обробки природної мови (NLP) для аналізу тональності новинних статей. Це дозволяє визначити, чи є новина позитивною, негативною або нейтральною. Для цього використовуються моделі машинного навчання, такі як:

- BERT (Bidirectional Encoder Representations from Transformers): Модель, що дозволяє аналізувати контекст тексту і визначати його тональність з високою точністю.

Google News використовує алгоритми машинного навчання для персоналізації подачі новин. Це означає, що кожен користувач бачить новини, які найбільше відповідають його інтересам і вподобанням. Використовуються такі методи:

- Колаборативна фільтрація: Метод, що базується на аналізі вподобань користувачів із подібними інтересами.

- Контентно-орієнтована фільтрація: Метод, що використовує інформацію про самі новини для формування рекомендацій.

Google News активно працює над виявленням фейкових новин за допомогою машинного навчання. Це включає аналіз тексту, метаданих та інших факторів новинних статей. Для цього використовуються такі методи:

- Алгоритми класифікації: Наприклад, логістична регресія та градієнтний бустинг, які допомагають оцінити ймовірність того, що новина є фейковою.
- Моделі глибокого навчання: Використовуються для більш складного аналізу тексту та метаданих.

Google News використовує машинне навчання для аналізу заголовків та змісту новинних статей, що дозволяє виявляти ключові теми та тренди. Це включає:

- Виділення сутностей (Named Entity Recognition, NER): Алгоритми виділяють і класифікують важливі сутності в тексті, такі як імена людей, організації, місця тощо.
- Аналіз ключових слів та фраз: Використання алгоритмів для виділення найважливіших слів і фраз у тексті, що допомагає зрозуміти основний зміст новини.

Впровадження машинного навчання дозволило Google News значно підвищити ефективність та точність аналізу новинного контенту. Основні переваги включають:

- Швидкість обробки: Алгоритми машинного навчання дозволяють швидко обробляти великі обсяги новинного контенту, забезпечуючи актуальність інформації.
- Точність: Використання сучасних методів машинного навчання, таких як глибокі нейронні мережі та BERT, дозволяє досягти високої точності в класифікації та аналізі новин.

– Персоналізація: Рекомендаційні системи забезпечують користувачам доступ до релевантного контенту, підвищуючи їх задоволення від використання сайту.

Google News демонструє успішне впровадження технологій машинного навчання у свої процеси, що дозволяє підвищити якість та ефективність роботи з новинним контентом. Це приклад того, як машинне навчання може бути використане для розв'язання складних завдань у галузі медіа та інформаційних технологій.

Цей детальний огляд ілюструє, як новинний сайт може використовувати машинне навчання для покращення своїх сервісів і задоволення потреб користувачів.[20]

1.4 Переваги машинного навчання

Машинне навчання є потужним інструментом, що дозволяє автоматизувати обробку великих обсягів даних, аналізувати складні закономірності та робити точні прогнози. Проте, як і будь-яка технологія, машинне навчання має свої переваги та недоліки. У цьому розділі розглянемо основні плюси і мінуси використання машинного навчання у контексті аналізу новинного контенту.

Переваги використання машинного навчання:

– Машинне навчання дозволяє автоматизувати багато рутинних завдань, таких як класифікація новин, аналіз тональності, виявлення фейкових новин тощо. Це значно зменшує витрати часу та ресурсів на обробку інформації.
– Сучасні алгоритми машинного навчання, особливо глибокі нейронні мережі, демонструють високу точність у розпізнаванні патернів і класифікації даних. Це забезпечує більш надійні результати аналізу новинного контенту.

- Машинне навчання добре підходить для обробки великих обсягів даних. Воно здатне ефективно працювати з мільйонами новинних статей, що особливо важливо для великих новинних агрегаторів, таких як Google News або Reuters.
- Алгоритми машинного навчання дозволяють створювати персоналізовані рекомендації для користувачів, що підвищує їх задоволення від використання новинних сервісів. Це досягається за допомогою аналізу вподобань та поведінки користувачів.
- Машинне навчання дозволяє ефективно виявляти нові тренди та аномалії у новинному потоці. Це корисно для швидкого реагування на нові події та забезпечення актуальності інформації.[21]

1.5 Недоліки використання машинного навчання.

Основні недоліки:

- Для навчання ефективних моделей машинного навчання потрібні великі обсяги якісних даних. Збір та підготовка таких даних можуть бути складними та затратними за часом процесами.
- Якість результатів машинного навчання сильно залежить від якості вхідних даних. Неправильні, неповні або упереджені дані можуть призвести до неправильних висновків та рішень.
- Сучасні моделі машинного навчання можуть бути дуже складними та важкими для розуміння. Це може ускладнити їхню інтеграцію та налаштування, а також виявлення та виправлення помилок.
- Використання машинного навчання може викликати етичні та правові питання, пов'язані з конфіденційністю даних, можливими упередженнями алгоритмів та їх впливом на суспільство. Наприклад, неправильна класифікація новин може вплинути на громадську думку.
- Навчання та виконання моделей машинного навчання можуть вимагати значних обчислювальних ресурсів, що може призвести до високих витрат на апаратне забезпечення та електроенергію.

Машинне навчання має численні переваги, що роблять його потужним інструментом для аналізу новинного контенту. Воно дозволяє автоматизувати процеси, підвищити точність та ефективність аналізу, а також забезпечити персоналізацію сервісів. Проте, разом з цим, існують і певні недоліки, такі як необхідність великих обсягів даних, чутливість до їх якості, складність моделей, етичні та правові питання, а також високі обчислювальні витрати. Розуміння цих аспектів дозволяє краще оцінити можливості та обмеження машинного навчання і використовувати його максимально ефективно.[22]

1.6 Обгрунтування необхідності допрацювання та проведення додаткового дослідження .

Незважаючи на значний прогрес у використанні машинного навчання для аналізу новинного контенту, існують певні області, які потребують додаткового дослідження та вдосконалення. У цьому розділі розглянемо основні проблеми та виклики, які вимагають доопрацювання, щоб підвищити ефективність та точність аналізу новинних сайтів. [23]

Однією з головних проблем у машинному навчанні є залежність від якості вхідних даних. Неправильні, неповні або упереджені дані можуть призвести до неправильних висновків. Для забезпечення високої точності моделей необхідно розробляти методи очищення та попередньої обробки даних, які дозволять зменшити вплив некоректних даних. (рис. 2.1).

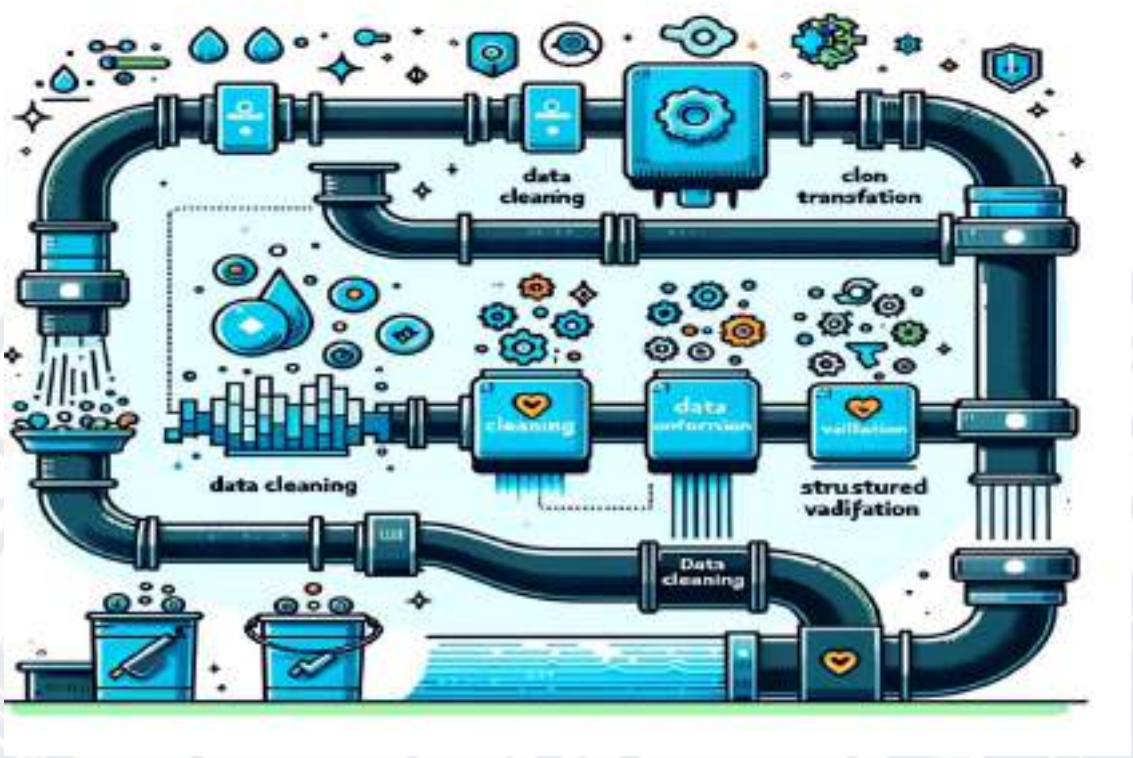

Рис. 2.1 - Процес очищення даних у машинному навчанні

Існуючі моделі машинного навчання можуть мати труднощі з адаптацією до нових тем і трендів, що з'являються у новинному потоці. Необхідно розробляти алгоритми, здатні швидко навчатися на нових даних і виявляти нові теми та тренди без значного зниження точності.

Більшість сучасних моделей машинного навчання розроблені для аналізу тексту на англійській мові. Аналіз новин на інших мовах потребує врахування специфіки кожної мови, що може ускладнювати процес обробки природної мови. Розробка багатомовних моделей або адаптація існуючих моделей до нових мов є важливим завданням.

Використання машинного навчання для аналізу новин піднімає питання етики та конфіденційності даних. Наприклад, можливі упередження моделей можуть призвести до дискримінації або неправильної класифікації новин. Необхідно розробляти методи для виявлення та зменшення упереджень у моделях машинного навчання.

У сучасному світі новини розповсюджуються з величезною швидкістю, тому важливо забезпечити обробку та аналіз даних у реальному часі. Це вимагає оптимізації алгоритмів машинного навчання для забезпечення високої продуктивності та низьких затримок у процесі обробки даних.

Потреби в нових методах та інструментах:

- Необхідно розробляти нові методи NLP, які дозволять більш точно аналізувати текст новин, враховуючи специфіку різних мов та контекстів. Це включає вдосконалення існуючих моделей, таких як BERT, та розробку нових архітектур, які забезпечують вищу точність та швидкість обробки.
- Розробка алгоритмів, здатних адаптуватися до нових даних і трендів у реальному часі, є важливим завданням. Це дозволить забезпечити актуальність та точність аналізу новин навіть у швидкозмінному інформаційному середовищі.
- Розробка методів для виявлення та зменшення упереджень у моделях машинного навчання допоможе забезпечити справедливість та етичність аналізу новин. Це може включати використання алгоритмів, які враховують різноманітні аспекти даних, та проведення регулярних перевірок моделей на упередження.
- Для забезпечення високої продуктивності та низьких затримок необхідно оптимізувати алгоритми машинного навчання для роботи на сучасних обчислювальних платформах. Це включає використання розподілених обчислень, GPU та інших технологій, що дозволяють обробляти великі обсяги даних у реальному часі.

Необхідність доопрацювання існуючих методів машинного навчання для аналізу новинного контенту є очевидною. Вирішення проблем, пов'язаних з якістю даних, адаптацією до нових тем, мовними бар'єрами, етичними питаннями та реальним часом обробки, дозволить значно підвищити ефективність та точність аналізу. Розробка нових методів та інструментів, таких як вдосконалені NLP моделі, алгоритми адаптивного навчання, техніки боротьби з упередженнями та

оптимізація обчислювальних ресурсів, є ключовими напрямами для подальшого розвитку цієї галузі. [27]

1.7 Можливі проблеми під час розробки та дослідження .

Під час проведення дослідження методів аналізу новинних сайтів на основі машинного навчання можуть виникати різні проблеми, які впливають на точність, ефективність та надійність результатів. У цьому розділі розглянемо основні виклики та проблеми, з якими можна зіткнутися під час дослідження, а також можливі шляхи їх вирішення.

Однією з основних проблем є неякісні та неповні дані. Сирі дані можуть містити помилки, пропущені значення або бути неповними, що може призвести до некоректних результатів машинного навчання.

- Рішення: Використання методів очищення та попередньої обробки даних, таких як видалення або заміна пропущених значень, корекція помилок та нормалізація даних.

Дослідження можуть бути піддані упередженості даних, коли наявні дані не представляють повну картину або мають певні схильності, що може призвести до упереджених моделей та неправильних висновків.

- Рішення: Проведення аналізу на предмет упередженості та використання методів для виявлення та зменшення впливу упередженості, таких як збалансування даних та розширення вибірки.

Обробка великих обсягів новинного контенту може вимагати значних обчислювальних ресурсів, що може сповільнити процес дослідження та аналізу.

- Рішення: Використання хмарних обчислень та розподілених обчислювальних платформ, які дозволяють ефективно масштабувати обчислювальні ресурси для обробки великих обсягів даних.

Сучасні моделі машинного навчання можуть бути дуже складними та важкими для налаштування та оптимізації. Це може призвести до тривалого часу навчання та високих витрат на обчислення.

- Рішення: Оптимізація моделей та використання більш ефективних алгоритмів, таких як градієнтний бустинг, який забезпечує високу точність при меншій складності.

Багато моделей машинного навчання, особливо глибокі нейронні мережі, можуть працювати як "чорна скринька", тобто їхні внутрішні процеси важко зрозуміти та інтерпретувати.

- Рішення: Використання методів інтерпретованого машинного навчання, таких як LIME або SHAP, які дозволяють пояснювати рішення моделей та виявляти важливі ознаки.

Іноді моделі можуть давати непередбачувані або некоректні результати через особливості даних або архітектури моделі.

- Рішення: Проведення ретельного тестування моделей на різних наборах даних та використання методів перевірки, таких як крос-валідація, для забезпечення надійності результатів.

Під час аналізу новинного контенту може виникнути проблема збереження конфіденційності даних, особливо якщо дослідження включає персональні або чутливі дані. [9]

- Рішення: Дотримання принципів етики та законодавства щодо конфіденційності даних, таких як GDPR, а також використання методів анонімізації та захисту даних.

Упереджені моделі можуть призвести до дискримінаційних результатів або неправильної класифікації новин, що може мати негативний вплив на суспільство.

– Рішення: Регулярний моніторинг моделей на предмет упередженості та впровадження методів для виявлення та зменшення упередженості.

Під час проведення дослідження методів аналізу новинних сайтів на основі машинного навчання можуть виникати різні проблеми, які впливають на точність, ефективність та надійність результатів. Розуміння цих проблем та пошук ефективних рішень дозволяють забезпечити високу якість та надійність дослідження. Важливо враховувати проблеми з якістю даних, обчислювальними ресурсами, інтерпретацією результатів та етичні й правові питання для досягнення успішних результатів.

РОЗДІЛ 2. ФОРМАЛІЗОВАНИЙ ОПИС МОДЕЛІ

2.1 Формування наукової гіпотези

У рамках проходження виробничої практики було поставлено завдання розробити новинний сайт із застосуванням машинного навчання для автоматизації аналізу та класифікації новинного контенту. Науковою гіпотезою цього дослідження є припущення, що впровадження моделей глибокого навчання, зокрема рекурентних нейронних мереж (RNN) та моделі BERT, дозволить суттєво підвищити точність класифікації новин, виявлення фейкових новин та персоналізації контенту.

Основні елементи гіпотези:

- Використання обробки природної мови (NLP) для аналізу новинного тексту;
- Моделі глибокого навчання здатні ефективно аналізувати великі обсяги новинного контенту;
- Автоматизація процесу класифікації новин знижує ризики людських помилок та забезпечує більшу ефективність порівняно з ручною обробкою;
- Використання алгоритмів для виявлення фейкових новин сприятиме підвищенню достовірності контенту.

2.2 Системний аналіз процесів.

Для вирішення завдань даного дослідження було обрано системний підхід, що дозволяє аналізувати новинний контент як багатовимірну систему. Основні етапи системного аналізу включають визначення вхідних параметрів, опис моделі обробки та отримання вихідних результатів.

Вхідні параметри системи:

- Новинні статті: текстовий контент, що підлягає аналізу, включаючи заголовки, авторів, час публікації.
- Лінгвістичні ознаки: частота вживання ключових слів, аналіз тональності, синтаксичні та семантичні структури тексту, кількість кліків,

– Метадані: інформація про джерело новини, тематику, популярність у користувачів.

Вихідні метрики (індикатори):

– Класифікація новин за категоріями.
– Індикатор тональності новини: позитивна, негативна або нейтральна.
– Рекомендаційні алгоритми: персоналізовані новини для кожного користувача.

2.3 Формалізація моделі дослідження.

У сучасному інформаційному просторі користувачі стикаються з величезною кількістю новинного контенту. Завдання полягає в тому, щоб розробити систему рекомендацій, яка персоналізовано підбиратиме статті для кожного користувача, базуючись на їхніх попередніх взаємодіях та характеристиках статей.

Наукова гіпотеза: Використання моделі машинного навчання для прогнозування ймовірності взаємодії користувача зі статтею на основі їхніх ознак дозволить покращити точність рекомендацій та підвищити залученість користувачів.

2.3.1 Параметри моделі та вхідні дані

Користувачі (U):

– Ідентифікатори користувачів u ∈ U.
– Можливі додаткові ознаки користувачів (демографічні дані, уподобання).

Статті (S):

– Категорія статті ($c \in C$) : тематичний розділ, до якого належить стаття.
– Тональність статті (t ∈ T): емоційне забарвлення (позитивна, нейтральна, негативна).
– Інші ознаки: довжина статті, дата публікації, аватар, тощо.

Взаємодії (I):

– Пара пар (користувач, стаття), де фіксується факт взаємодії (перегляд, оцінка).

2.3.2 Формалізація задачі

Модель рекомендаційної системи можна представити як задачу бінарної класифікації, де цільова змінна y визначає, чи буде користувач взаємодіяти зі статтею.

- **Вхідний вектор ознак (X) для кожної пари (користувач, стаття):**

$$X = [x_1, x_2, x_3, ..., x_n]$$

Де:

- x_1: кодований ідентифікатор користувача.
- x_2: кодована категорія статті.
- x_3: кодована тональність статті.
- x_i: додаткові числові або кодовані ознаки.

- **Цільова змінна (y):**

$$y = \begin{cases} 1, & \text{якщо користувач взаємодіяв зі статтею} \\ 0, & \text{якщо взаємодії не було} \end{cases}$$

Рисунок 2.3.2 – Вхідний вектор ознак

2.3.3 Алгоритм моделі

Для роботи моделі машинного навчання необхідно перетворити категоріальні ознаки (такі як категорія та тональність статті) у числовий формат. Це здійснюється за допомогою методу кодування міток, який присвоює унікальні числові значення кожній категорії.[30]

Кодування ознак:

– Використання LabelEncoder для перетворення категоріальних ознак у числові значення:

$$\text{user\_encoded} = \text{LabelEncoder}(\text{user\_id})$$

$$\text{category\_encoded} = \text{LabelEncoder}(\text{category})$$

$$\text{tone\_encoded} = \text{LabelEncoder}(\text{tone})$$

Рисунок 2.3.2 – Вектор ознак

Створення вибірки:

- Позитивні приклади: пари (користувач, стаття), де відбулася взаємодія(y=1).
- Негативні приклади: випадково вибрані пари (користувач, стаття), де взаємодії не було (y=0).

2.3.4 Вибір та обґрунтування моделі

Для вирішення поставленої задачі обрано модель випадкового лісу (Random Forest Classifier). Ця модель є потужним інструментом для задач класифікації, оскільки:

- Добре працює з різними типами ознак і не вимагає масштабування даних.
- Стійка до перенавчання завдяки використанню ансамблю дерев рішень.
- Здатна обробляти нелінійні залежності між ознаками.

2.3.5 Процес навчання моделі

Етап 1: Підготовка даних

Після кодування ознак формується навчальна та тестова вибірки. Навчальна вибірка використовується для побудови моделі, а тестова — для оцінки її якості.

Етап 2: Навчання моделі

Модель випадкового лісу навчається на навчальній вибірці, аналізуючи взаємозв'язки між ознаками користувача та статті та фактом взаємодії.

Етап 3: Оцінка моделі

На тестовій вибірці оцінюється точність моделі та інші метрики, що дозволяють судити про її ефективність.

2.3.5 Алгоритм генерації рекомендацій

Крок 1: Для кожного користувача визначається список статей, з якими він ще не взаємодіяв.

Крок 2: Для цих статей формується набір ознак, який використовується моделлю для прогнозування ймовірності взаємодії.

Крок 3: Модель прогнозує ймовірність того, що користувач зацікавиться кожною статтею.

Крок 4: Статті сортуються за зменшенням прогнозованої ймовірності, і користувачу рекомендуються ті, що мають найвищі значення.

2.3.6 Алгоритм генерації рекомендацій

Вплив ознак на рекомендації:

- Категорія статті: Може свідчити про інтереси користувача в певних темах.
- Тональність статті: Деякі користувачі можуть віддавати перевагу статтям з певним емоційним забарвленням.

Аналіз важливості ознак:

- Модель випадкового лісу дозволяє оцінити, які ознаки найбільше впливають на прийняття рішення. Це допомагає зрозуміти, які фактори є ключовими при рекомендації контенту.

2.3.6 Масштабованість та адаптивність системи

Обробка нових користувачів та статей:

- Система може стикатися з ситуацією, коли з'являються нові користувачі або статті. Для нових користувачів, які ще не мають історії взаємодій, можуть рекомендуватися найпопулярніші статті або ті, що відповідають загальним трендам.

Оновлення моделі:

- Регулярне перенавчання моделі дозволяє враховувати нові дані та адаптуватися до змін у поведінці користувачів.

2.3.7 Практичне застосування та результати

Впроваджена модель рекомендацій дозволяє підвищити релевантність пропонованого контенту для користувачів, що сприяє збільшенню їхньої

залученості та часу, проведеного на платформі. Попередні тести показали, що модель ефективно прогнозує інтерес користувачів до статей, базуючись на аналізі їхніх взаємодій та характеристик контенту.[33]

2.4 Опис методики дослідження

Основні етапи дослідження:

- Збір та підготовка даних: На першому етапі було здійснено збір новинних текстів з різних джерел. Ці дані були очищені від шуму, тобто зайвих символів, дублікатів та недостовірної інформації.
- Обробка природної мови (NLP): Використовувалися такі методи, як токенізація, лематизація, виділення сутностей та аналіз тональності. Це дало змогу підготувати тексти для подальшого машинного навчання.
- Навчання моделей: На етапі навчання використовувалися моделі глибокого навчання для класифікації тексту та аналізу тональності. Для підвищення точності моделей застосовувалися сучасні методи регуляризації, такі як дропаут.
- Аналіз отриманих результатів: На виході система генерує метрики точності класифікації, які оцінюються через крос-валідацію та інші методи оцінки.

2.5 Програмно апаратний інструментарій

Для побудови сайту новин було використано наступні засоби: Python, Django, HTML, CSS, JavaScript, Bootstrap, TextBlob, SQLite.

Python — це інтерпретована мова програмування високого рівня, відома своєю простотою, читабельністю та універсальністю. Він був розроблений Гвідо ван Россумом і вперше випущений у 1991 році. Філософія дизайну Python наголошує на чіткості та лаконічності коду, що полегшує його написання, читання та підтримку.Веб-сайт, розроблений в рамках дипломної роботи с

Однією з визначних особливостей Python є його чіткий і читабельний синтаксис. Він використовує пробільні відступи для визначення блоків коду, що покращує читабельність коду та сприяє послідовному форматуванню. Цей

вибір дизайну також усуває потребу в явних дужках або роздільниках, що робить код Python візуально чистим.

Python підтримує кілька парадигм програмування, зосереджуючись на об’єктно-орієнтованому програмуванні (ООП). Це дозволяє розробникам визначати класи, створювати об’єкти та впроваджувати успадкування, поліморфізм та інкапсуляцію. Однак Python не обмежується ООП, а також підтримує процедурні та функціональні стилі програмування.

Django — це дуже популярна та потужна веб-платформа, написана на Python, яка спрощує та оптимізує процес розробки веб-додатків. Він надає повний набір інструментів, бібліотек і функцій, розроблених, щоб допомогти розробникам ефективно створювати надійні, масштабовані та безпечні веб-додатки..

Велика екосистема Django ще більше розширює його можливості. Він пропонує велику колекцію багаторазових пакетів і бібліотек через індекс пакетів Python (PyPI). Ці пакети охоплюють різні домени, такі як підключення до бази даних, кешування, обробка електронної пошти, REST API тощо, що дозволяє розробникам використовувати існуючі рішення та прискорювати розробку.

Загалом сильні сторони Django полягають у його здатності спрощувати й автоматизувати звичайні завдання веб-розробки, його зосередженості на повторному використанні коду та зручності обслуговування, а також у його наголосі на безпеці та масштабованості. Він набув широкого поширення в галузі та продовжує залишатися популярним вибором для розробки всіх типів веб-додатків, від невеликих проектів до великих корпоративних систем.

HTML (мова гіпертекстової розмітки) і CSS (каскадні таблиці стилів) — дві основні технології, які використовуються для створення та оформлення веб-сторінок. Вони працюють разом, щоб структурувати та подавати вміст у мережі.

HTML є стандартною мовою розмітки для створення веб-сторінок. Він забезпечує структурований формат для організації вмісту та визначення його

семантичного значення. HTML складається з набору елементів або тегів, які використовуються для розмітки різних частин веб-сторінки. Ці елементи описують структуру та ієрархію вмісту, наприклад заголовки, абзаци, списки, зображення, таблиці, форми тощо.

CSS — це мова таблиць стилів, яка використовується для опису представлення та стилю документів HTML. Він визначає, як елементи мають відображатися на веб-сторінці, включаючи їхній макет, кольори, шрифти, інтервали тощо. CSS працює, орієнтуючись на елементи HTML і застосовуючи до них правила стилю.

CSS також підтримує концепцію каскаду, коли до елемента можна застосувати кілька правил стилю. У випадках суперечливих правил CSS дотримується певного порядку пріоритету для визначення остаточного стилю.

HTML і CSS працюють рука об руку, створюючи візуально привабливі та структуровані веб-сторінки. HTML визначає вміст і структуру сторінки, тоді як CSS додає презентацію та візуальний стиль. Разом вони утворюють основу веб-розробки, дозволяючи розробникам створювати цікаві та зручні веб-сайти

JavaScript — це динамічна та універсальна мова програмування, яка відіграє вирішальну роль у веб-розробці. Він був розроблений для забезпечення інтерактивності та покращення взаємодії з веб-сторінками. JavaScript переважно використовується на стороні клієнта, але з появою Node.js він розширився до розробки на стороні сервера.

Однією з ключових особливостей JavaScript є його здатність маніпулювати об’єктною моделлю документа (DOM) веб-сторінок і взаємодіяти з нею. Це дозволяє розробникам динамічно змінювати елементи HTML, оновлювати вміст, обробляти події та створювати інтерактивні інтерфейси користувача. JavaScript може реагувати на дії користувача, такі як натискання кнопок, надсилання форм і введення з клавіатури, прикріплюючи прослуховувачі подій до елементів DOM.

JavaScript підтримує різні парадигми програмування, включаючи імперативне, об’єктно-орієнтоване та функціональне програмування. Він надає

такі функції, як змінні, типи даних, оператори, оператори потоку керування та функції. Змінні в JavaScript динамічно типізуються, тобто вони можуть містити значення різних типів даних. Функції в JavaScript розглядаються як об’єкти першого класу, що дозволяє призначати їх змінним, передавати як аргументи та повертати як значення. Ця гнучкість дає змогу розробникам писати код у різних стилях і підходах.

Bootstrap — це широко поширена інтерфейсна платформа з відкритим вихідним кодом, яка спрощує процес проектування та розробки адаптивних і орієнтованих на мобільні пристрої веб-сторінок і програм. Він надає колекцію попередньо створених стилів CSS, компонентів JavaScript і адаптивних утиліт макета, які допомагають розробникам створювати візуально привабливі та узгоджені веб-інтерфейси.

Однією з ключових особливостей Bootstrap є його адаптивний дизайн. Фреймворк включає адаптивну систему сітки, що складається з 12-стовпцевого макета, який автоматично регулює позиціонування та розмір елементів на основі розміру екрана та типу пристрою. Це гарантує, що веб-сторінки, створені за допомогою Bootstrap, оптимізовані для різних пристроїв, включаючи настільні ПК, планшети та мобільні телефони.

TextBlob — це бібліотека Python, створена на основі набору інструментів природної мови (NLTK), яка забезпечує простий та інтуїтивно зрозумілий API для поширених завдань обробки природної мови (NLP). Це дозволяє розробникам аналізувати та обробляти текстові дані, отримувати значущу інформацію та виконувати такі завдання, як аналіз настроїв, додавання тегів до частин мови, виділення іменників, переклад мови тощо.

Однією з ключових особливостей TextBlob є простота використання. Він забезпечує простий і високорівневий інтерфейс, який абстрагує складність НЛП, роблячи його доступним навіть для тих, хто не має глибоких знань або досвіду НЛП. API TextBlob розроблений таким чином, щоб бути зручним для користувача, з інтуїтивно зрозумілими методами та властивостями, які

дозволяють розробникам виконувати різні завдання NLP з мінімальною кількістю коду.

SQLite — це популярна й широко використовувана система керування реляційною базою даних (RDBMS), яка відома своєю простотою, гнучкістю та легкістю. На відміну від традиційних систем баз даних клієнт-сервер, SQLite працює як безсерверна файлова база даних, яка вбудована безпосередньо в програми.

Однією з ключових особливостей SQLite є його автономна та безсерверна архітектура. Для цього не потрібен окремий сервер бази даних або процес встановлення, оскільки вся база даних міститься в одному файлі в локальній файловій системі. Це робить SQLite дуже портативним і простим у розгортанні, оскільки його можна поширювати як один файл разом із програмою.[1]

2.6 ER-діаграма та Схема IDEF 0

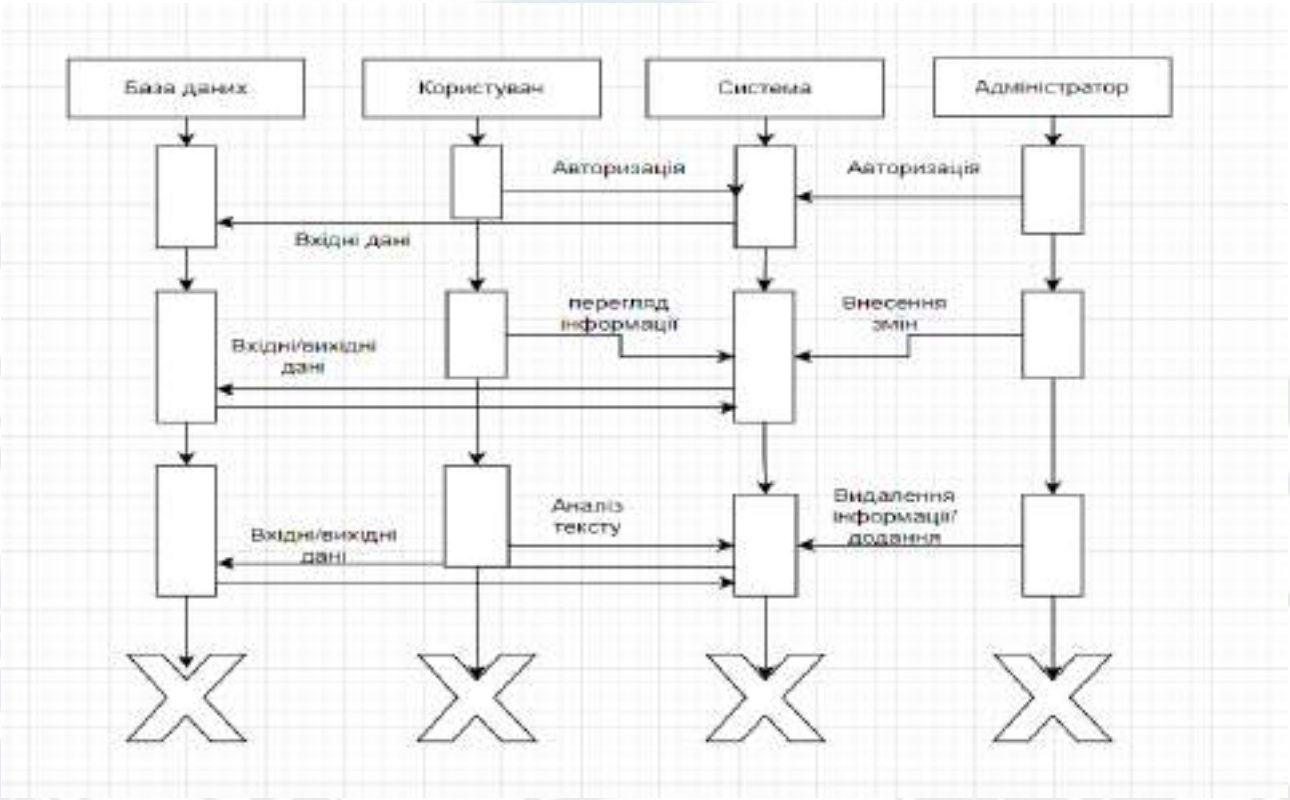

Рисунок 2.6.1 – ER-діаграма

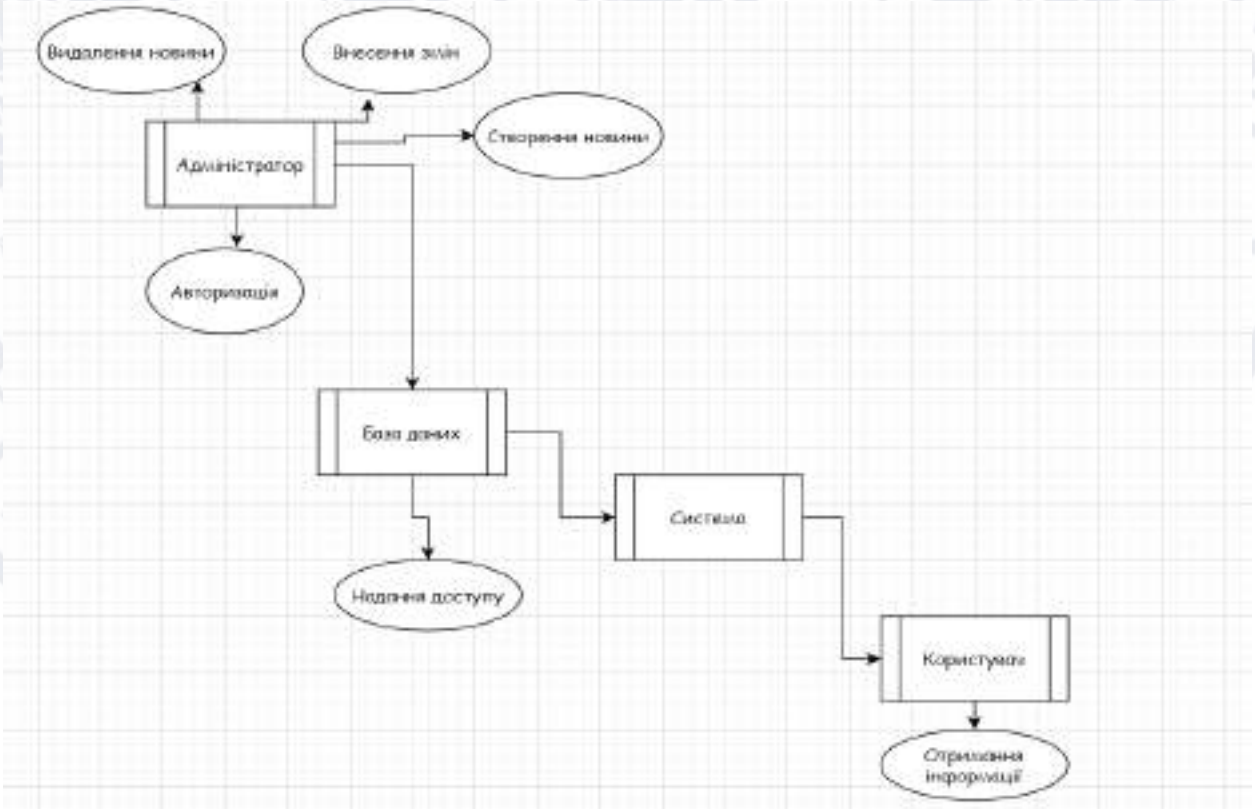

Рисунок 2.6.2 – Схема IDEF 0

2.7 Опис вхідних/вихідних даних

Вхідні, проміжні та вихідні дані відіграють ключову роль у процесах обробки та аналізу інформації в обчислювальних системах. Кожен із цих типів даних має своє призначення і є основою для безперебійного функціонування різних програмних процесів.

Вхідні дані — це інформація, яка подається в систему, процес або модель для подальшої обробки. Типи вхідних даних можуть суттєво варіюватися залежно від конкретного застосування. Наприклад, для моделей машинного навчання вхідними даними може бути набір даних, що містить інформацію для навчання моделі. У випадку новинного веб-сайту вхідними даними можуть бути запити користувачів, кліки або сирий текст новинних статей, які необхідно обробити та проаналізувати.

Проміжні дані — це інформація, яка генерується та використовується в системі під час обробки, але не є кінцевими результатами. Вони часто слугують етапами на шляху до досягнення остаточного результату. Наприклад, у машинному навчанні проміжні дані можуть включати різні метрики або параметри, які оновлюються під час навчання моделі. На новинному веб-сайті проміжні дані можуть бути аналізованим та токенізованим текстом статей або даними про поведінку користувачів, які були частково оброблені, але ще не завершені.

Вихідні дані — це остаточні результати, які система генерує після завершення обробки. У моделях машинного навчання вихідними даними можуть бути прогнози, зроблені моделлю на нових даних, або остаточні ваги натренованої моделі. Для новинного веб-сайту вихідними даними можуть бути результати пошукових запитів, які повертаються користувачам, або результати аналізу настроїв щодо конкретних статей.

Опис цих типів даних є досить загальним, оскільки їхня конкретна природа може значно відрізнятися в залежності від контексту застосування. Проте, незалежно від специфіки, розуміння ролі вхідних, проміжних та вихідних даних

є надзвичайно важливим для ефективного проектування та впровадження систем обробки даних або моделей машинного навчання.[2]

2.8 Вимоги до ПЗ та АЗ, що необхідно для виконання програми

Вимоги до програмного забезпечення:

- Операційна система Windows 8, 64-розрядна або вище.

Вимоги до апаратного забезпечення:

- Процесор – 1 ГГц або більше;
- Оперативна пам’ять 128 мб або більше;
- Комп’ютерна «миша» та клавіатура/смартфон;
- ОС Windows;
- Жорсткий диск 128 мб або більше.

РОЗДІЛ 3. ОГЛЯД САЙТУ ТА МОЖЛИВОСТІ ДЛЯ ПОДАЛЬШОГО ДОСЛІДЖЕННЯ

3.1 Функціональність та користувацький інтерфейс

Після запуску локального серверу, відбувається запуск веб-додатку. При цьому рендериться сторінка з інтерфейсом, що зображено на рисунку – 3.1.1.

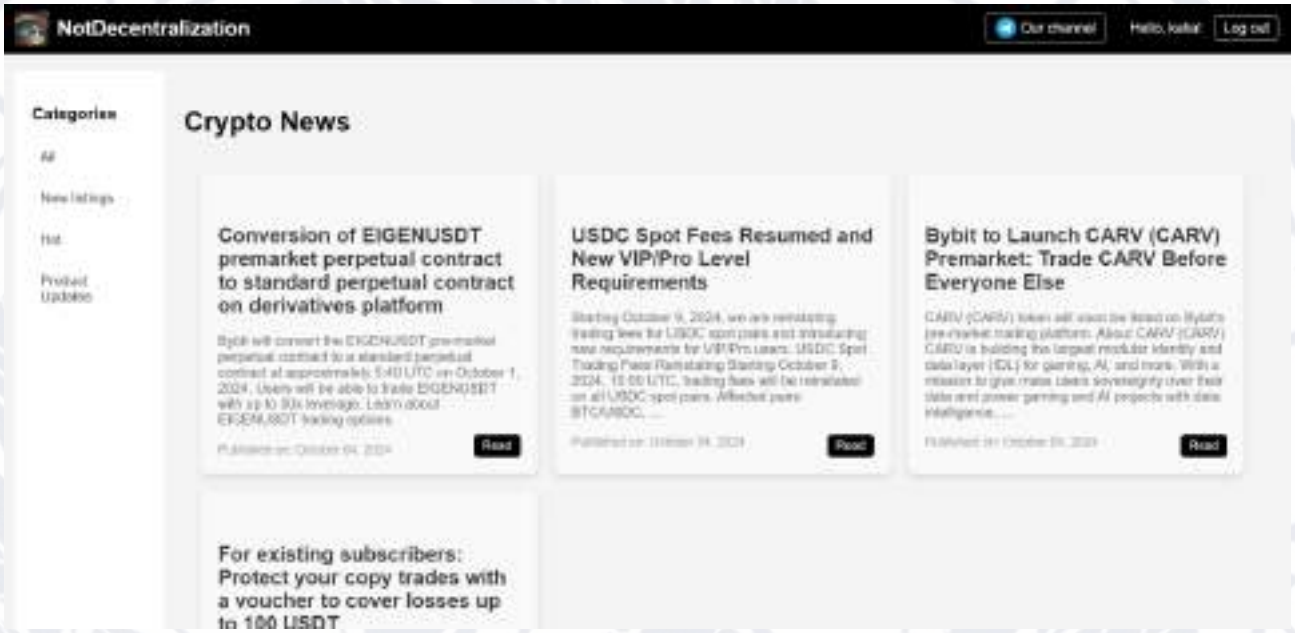

Рисунок 3.1.1 – Головна сторінка

Огляд усіх функціональних можливостей користувача з адміністратором та розміщення їх трігерів на інтерфейсі:

- На сторінці «News», що зображено на рисунку 3.1.1. меню завдяки якому користувач може переходити на потрібну йому сторінку обравши відповідний розділ меню. Натискаючи на кожну з кнопок користувач переходить на наступну сторінку.
- На сторінці «New listing», що зображено на рисунку 3.1.2. відображаються всі новини на сайті пов’язані із листінгом криптовалют, які має можливість вносити адміністратор Адміністратор має можливість вносити інформацію через панель адміністратора.

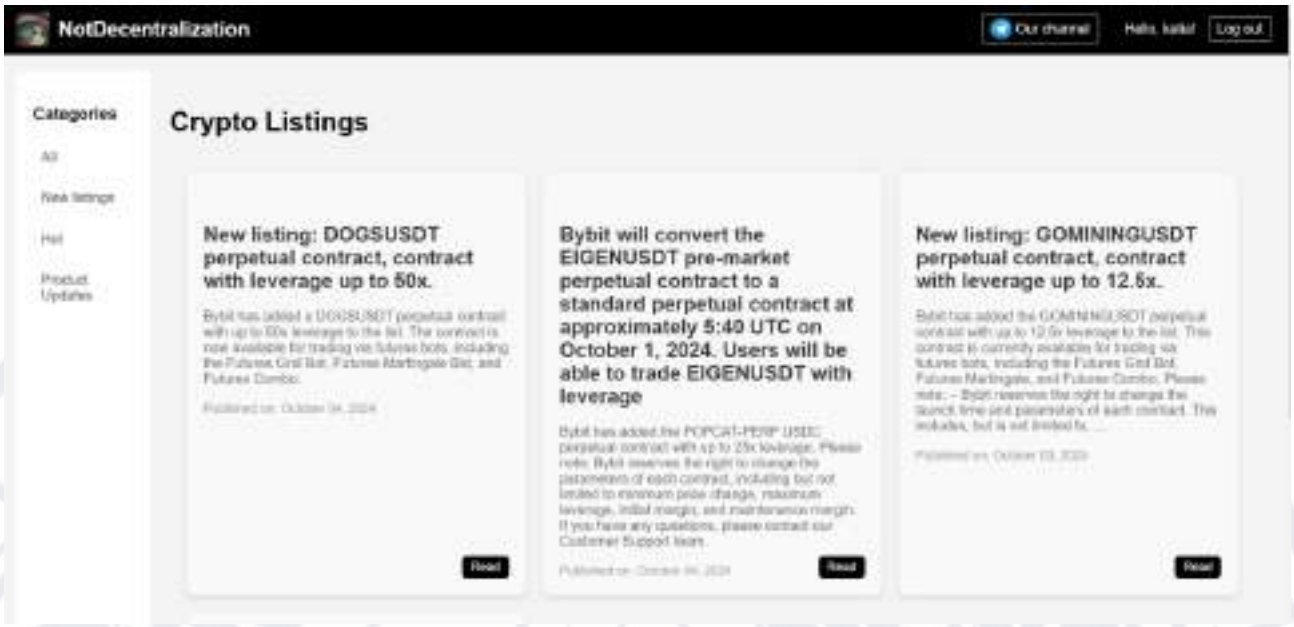

Рисунок 3.1.2 – Сторінка «Список новин»

- На сторінці «HOT», що зображено на рисунку 3.1.3.

відображаються список гарячих новин, які додаються через панель адміністратора.

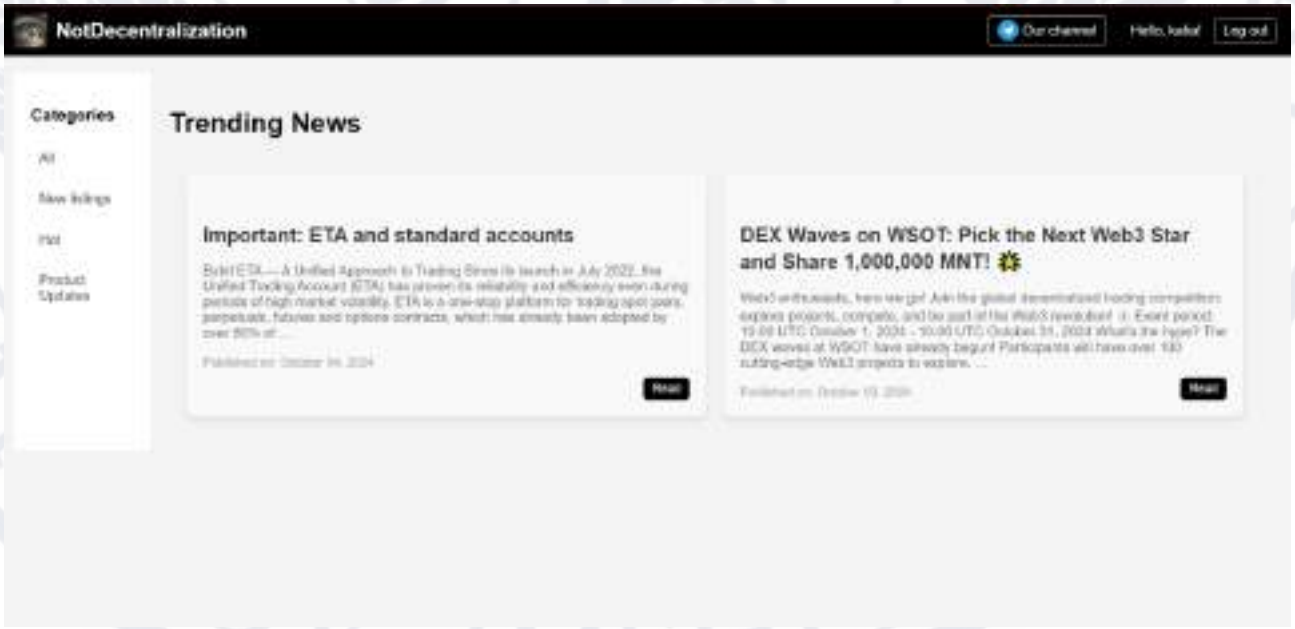

Рисунок 3.1.2 – Сторінка «Гаряче»

- На сторінці «Latest News», що зображено на рисунку 3.1.4.

відображаються популярних новин, які додаються через панель адміністратора.

Користувач має можливість натиснути на опис новини та прочитати детальний опис новини.

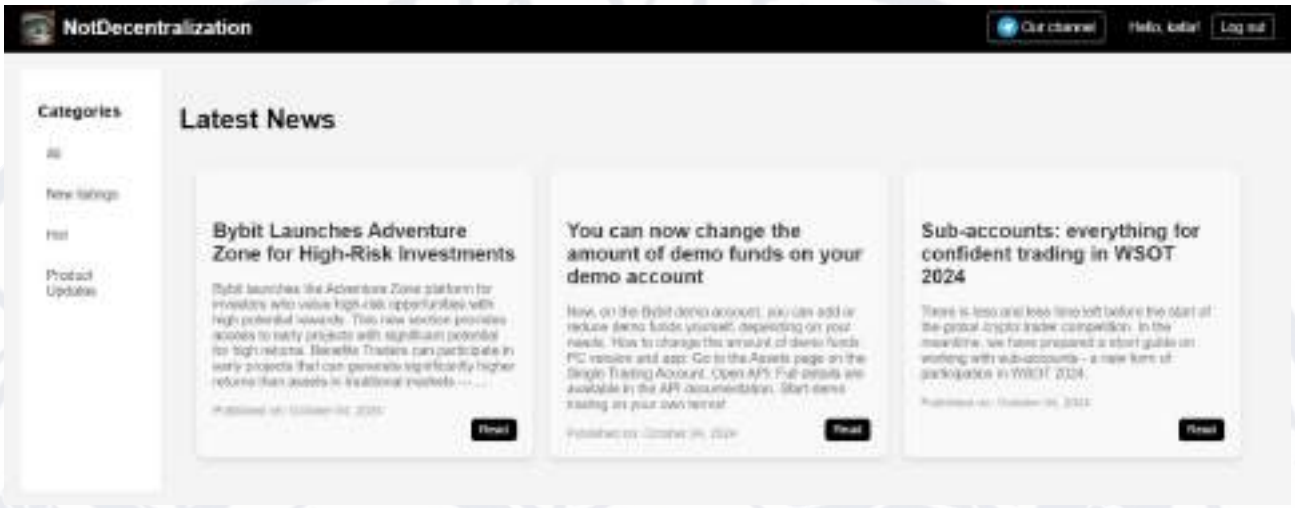

Рисунок 3.1.4 – Сторінка «Популярне»

- На кожній сторінці в змісті новини реалізована кнопка «Читати більше». Користувач має можливість натиснути на цю кнопку та перейти на сторінку з детальним описом новини, що зображено на рисунку 3.1.7 , також кожен користувач отримує індивідуальні рекомендації новин, но основі його дії та інтересів 3.1.8.

Рисунок 3.1.7 – Сторінка «Читати більше»

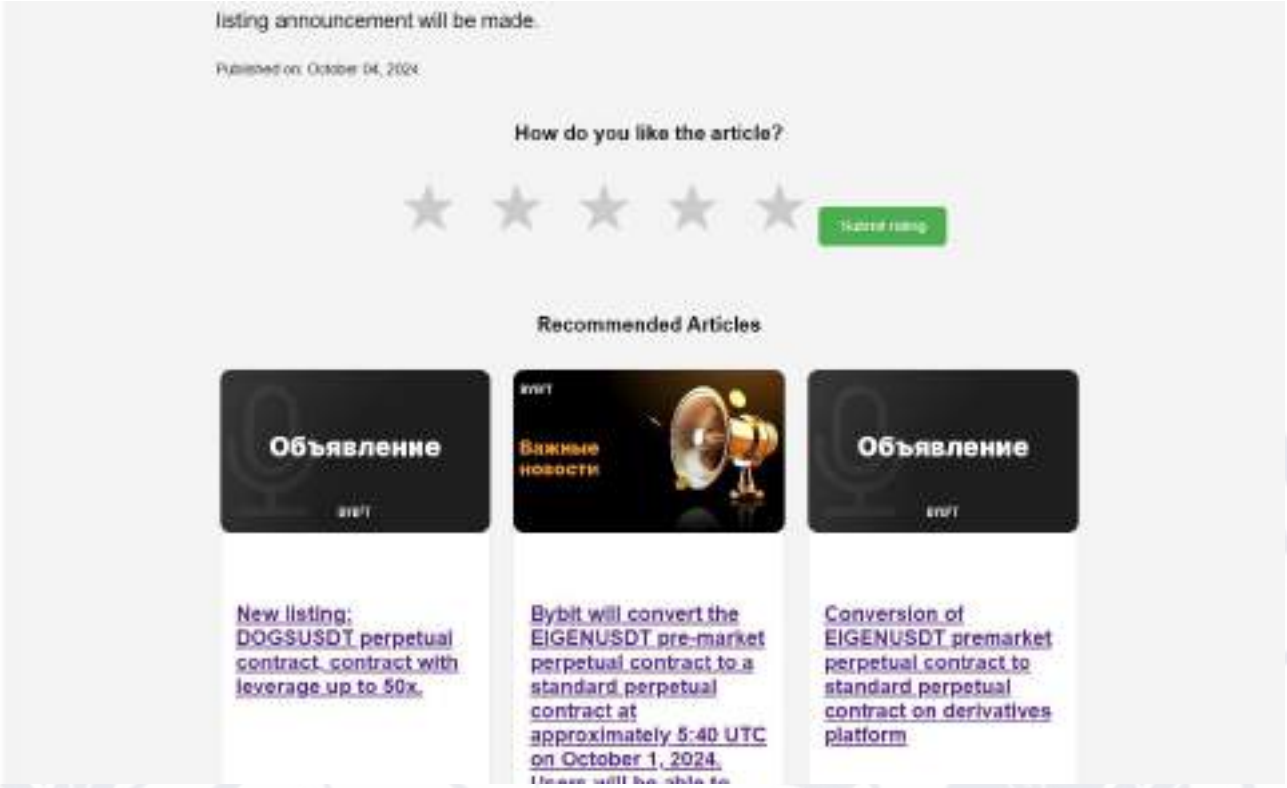

Рисунок 3.1.8 – Сторінка «Читати більше з персональними рекомендаціями»

- На сторінці авторизації в панелі адміністратора , що зображено на рисунку 3.1.9. є два поля логін та пароль. Адміністратор здійснює авторизацію в панелі адміністратора за допомогою вже створеного логіну та парою.

Рисунок 3.1.9 – Сторінка «Авторизація в панелі адміністратора»

- В панелі адміністратора, що зображено на рисунку 3.1.10. є три розділи. Адміністратор має можливість переглядати дії користувачів, додавати/видаляти новини, переглядати список користувачів.

Рисунок 3.1.10 – Сторінка «Панель адміністратора»

- В розділі «Articles» , що зображено на рисунку 3.1.11 адміністратор має можливість додати нову новину. Для цього потрібно задати заголовок новини, опис та завантажити потрібне зображення. Також, адміністратор має можливість обрати на якій сторінці сайту буде публікуватись ця новина.

Рисунок 3.1.11 – Сторінка «Панель адміністратора»

- Користувачі можуть зареєструватись на сайті через кнопку реєстрації, що зображено на рисунку 3.1.12.

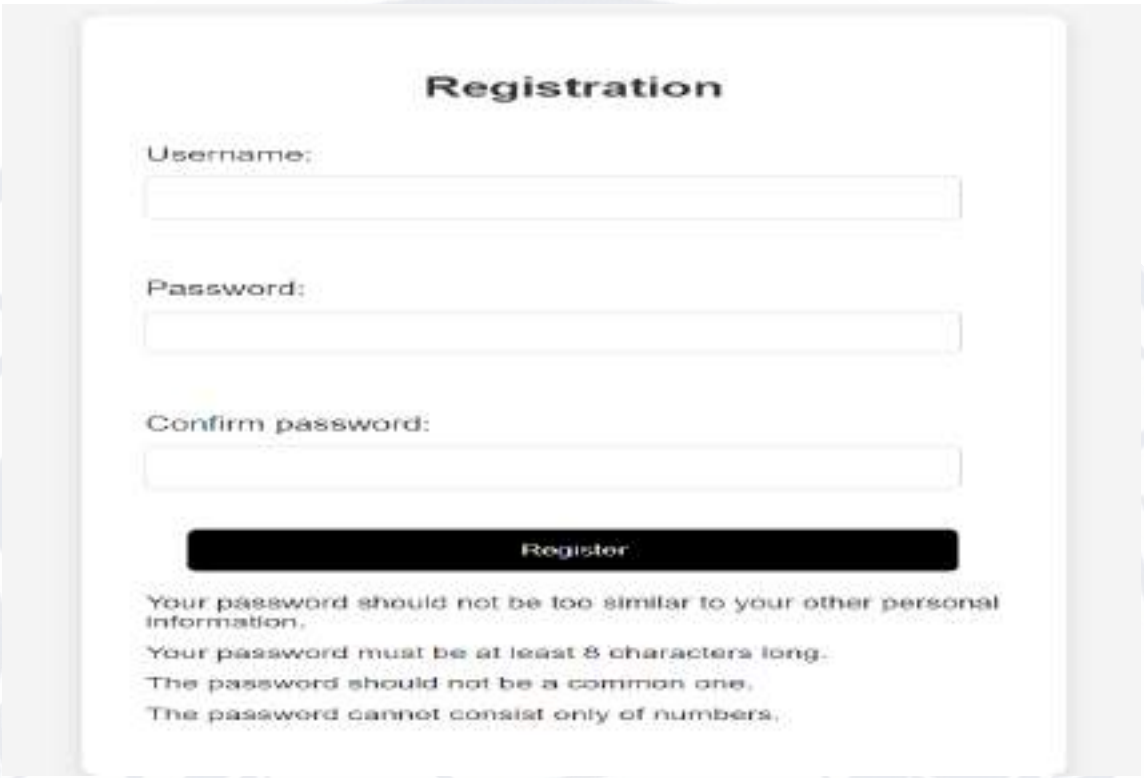

Рисунок 3.1.12 – Сторінка «Реєстрації»

- Рекомендація новин відбувається виключно для авторизованих користувачів спеціально навченою моделлю яка тренується на діях користувача та рекомендує новини.

Рисунок 3.1.11 – Процес рекомендації новин у консолі

3.2 Опис тестування з екранними формам

Тестування включає в себе перевірку працездатності програми при різних вхідних даних та залежність програми від зовнішніх факторів, програмного чи апаратного засобів.

Вікно авторизації панелі адміністратора (вхід до кабінету), таблиця 3.2.1:

Таблиця 3.2.1. – Панель адміністратора

Тест	**Очікуваний результат**	**Фактичний результат**	**Результат тестування**
Вхід на сайт	Рендер сторінки	Успішний рендер сторінки	Успіх
Перевірка роутингу	Роутинг працює коректно, навігація по сайту відповідає умовам	Роутинг працює, навігація відповідає поставленим цілям	Успіх
Ввід даних	При натисканні на форми вводу, набирається текст з клавіатури	Можна набирати текст з клавіатури	Успіх
Перевірка даних, що вводяться	При натисканні на кнопку Enter відбувається перехід на особистий кабінет	Перехід на особистий кабінет	Успіх

Робота меню сайту, таблиця 3.2.2:

Тест	**Очікуваний результат**	**Фактичний результат**	**Результат тестування**
Кнопка «All»	Перехід на потрібну сторінку	Успішний перехід на потрібну сторінку	Успіх
Кнопка «New listing»	Перехід на потрібну сторінку	Успішний перехід на потрібну сторінку	Успіх

Кнопка «Hot»	Перехід на потрібну сторінку	Успішний перехід на потрібну сторінку	Успіх
Кнопка «Product Updates»	Перехід на потрібну сторінку	Успішний перехід на потрібну сторінку	Успіх

Робота сторінки з детальним описом новини та відповідних функцій на цій сторінці, таблиця 3.2.3:

Тест	**Очікуваний результат**	**Фактичний результат**	**Результат тестування**
Перехід на сторінку	Рендер сторінки	Успішний рендер сторінки	Успіх
Перевірка роутингу	Роутинг працює коректно, навігація по сайту відповідає умовам	Роутинг працює, навігація відповідає поставленим цілям	Успіх
Кнопка «Зареєструватись»	Реєстарція	Успішний перехід на головну сторінку сайту	Успіх
Кнопка «Увійти»	При правильному введенні логіну та паролю – авторизація в системі	Успішна авторизація	Успіх

3.3 Аналіз результатів тестування системи

Після проведення тестування сайту, ми отримали наступні результати:

1. Функціональне тестування: протестовано всі основні функції системи, включаючи вибірку та відображення статей новин, сортування та фільтрацію новин (наприклад, показ лише гарячих чи популярних новин), пошук новин за запитом, аналіз тексту, опублікованого в деталях новин.

перегляд та перегляд зображень банерів. Система успішно виконала всі ці завдання, що свідчить про правильну функціональну реалізацію.

2. Тестування продуктивності: система продемонструвала здатність обробляти значну кількість запитів одночасно. Навіть при великому навантаженні швидкість відповіді сервера та навантаження залишалися прийнятними, що свідчить про оптимальне використання ресурсів і відповідність вимогам до продуктивності.

3. Тестування сумісності: програма успішно працювала в різних браузерах і пристроях. Завдяки адаптивному дизайну Bootstrap веб-сайт добре адаптується до різних розмірів екрана. Це підтверджує адаптивність і сумісність системи..

4. Тестування безпеки: були застосовані відповідні методи безпеки Django. Django за своєю суттю забезпечує захист від кількох типових векторів атак. Будь-яка передача даних, авторизація користувачів і автентифікація повинні здійснюватися за допомогою сучасних протоколів безпеки.

5. Тестування зручності користування: з точки зору зручності використання та доступності програма пропонує інтуїтивно зрозумілу структуру навігації зі зрозумілим меню та кнопками, а також естетично привабливий дизайн. Ці фактори сприяють позитивній взаємодії з користувачем, вказуючи на те, що система відповідає вимогам щодо ергономіки та взаємодії з користувачем.

6. Тестування на помилки: Тестування системи спрямоване на виявлення будь-яких можливих помилок під час роботи. Додаток продемонстрував стабільність і відсутність критичних помилок, що впливають на роботу системи. Будь-які виявлені незначні помилки були виправлені та не вплинули на продуктивність системи.

7. Регресійне тестування: сайт продемонстрував стабільну роботу та відсутність нових дефектів, що підтверджує якість розробки та тестування.На основі аналізу результатів тестування, можна зробити висновок, що система

успішно пройшла всі етапи тестування та відповідає вимогам та очікуванням користувачів.

Підсумовуючи, виходячи з аналізу результатів тестування, система новинного сайту успішно пройшла всі етапи тестування та відповідає вимогам та очікуванням користувачів. Він готовий до розгортання та використання.

3.4 Проблеми виявлені під час дослідження та шляхи їх вирішення

Під час дослідження сайту новин було виявлено наступні проблеми:

1. Неточний аналіз: модель машинного навчання може не надавати точного аналізу. Це може бути наслідком поганого навчання моделі, неадекватних чи упереджених даних навчання або невідповідного вибору моделі для поточного завдання.
2. Низька продуктивність: якщо модель машинного навчання складна, а обчислювальні ресурси сервера обмежені, аналіз може зайняти багато часу, що призведе до поганої взаємодії з користувачем.
3. Переобладнання або недостатнє пристосування: якщо модель машинного навчання переобладнана, вона добре працюватиме з навчальними даними, але погано з новими, невідомими даними. Якщо він недостатньо підготовлений, він працюватиме погано навіть на тренувальних даних. Обидві ситуації можуть призвести до неточних аналізів.

Потенційні рішення та рекомендацій для вирішення зазначених проблем:

- Вибір моделі та налаштування: тестування різних моделей та налаштування гіперпараметрів. Використовуючи такі методи перевірки, як k-кратна перехресна перевірка, щоб переконатися, що модель добре узагальнюється.
- Використання методів ансамблю: поєднання кількох моделей часто може дати кращі результати, ніж використання однієї моделі.
- Оптимізація моделі: використання методів зменшення розмірності або простіші моделі, якщо це можливо, без значної втрати точності.

- Використання ефективної інфраструктури: використання більш потужних серверів та розподілення обчислення між кількома машинами. Хмарні рішення, такі як AWS, Google Cloud або Azure, надають масштабовані ресурси.
- Асинхронна обробка: якщо відповідь у реальному часі не потрібна, спробувати обробляти запити асинхронно та сповіщати користувача про завершення аналізу.
- Потрібно надати індикатор завантаження або панель прогресу, щоб повідомити користувачеві, що аналіз триває.
- Запровадити надійну обробку помилок і надати користувачеві чіткі повідомлення про помилки.
- Використання попередньо підготовлених моделей. Такі моделі, як BERT, RoBERTa або XLNet, попередньо навчені на великій кількості даних і можуть забезпечувати більш точний аналіз настроїв.

Головне пам’ятати, що машинне навчання – це ітеративний процес. Потрібно не боятись експериментувати, оцінювати та повторювати свої моделі та стратегії.

ВИСНОВОК

Підсумовуючи, впровадження методів машинного навчання в новинні веб-сайти має великий потенціал для покращення взаємодії з користувачами та глибшого аналізу контенту. Здатність аналізувати та класифікувати новинні статті, проводити аналіз настроїв і адаптуватися до вподобань користувачів може значно підвищити релевантність і персоналізацію контенту, що надається кожному відвідувачу.

Проте дослідження також виявили певні виклики, пов’язані з впровадженням машинного навчання на таких платформах. Забезпечення точності та послідовності аналізу, задоволення обчислювальних вимог для обробки даних у режимі реального часу та ефективна передача результатів користувачам – це аспекти, які потребують ретельного розгляду. Крім того, підтримання надійності при обробці непередбачених або некоректних вхідних даних, а також збереження ефективності та швидкодії є ще одними важливими чинниками, які слід враховувати.

Варто також зазначити, що використані методи машинного навчання, незважаючи на їхню передовість, не є безпомилковими. Можливі неточності в аналізі та класифікації, а також варіативність продуктивності залежно від конкретних завдань. Моделі машинного навчання потребують постійного навчання та вдосконалення для підтримки їхньої точності та актуальності.

На перспективу існує кілька обіцяючих напрямків для подальших досліджень. Наприклад, застосування більш складних методів обробки природної мови, таких як BERT або GPT-3, може забезпечити більш точний і деталізований аналіз новинного контенту. Крім того, дослідження використання глибокого навчання для аналізу настроїв і категоризації може привести до значних покращень.

Загалом, хоча інтеграція машинного навчання в новинні веб-сайти створює як можливості, так і виклики, це багатообіцяюча сфера розвитку, яка може революціонізувати спосіб взаємодії з новинним контентом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація Python [Онлайн]. Доступно: https://docs.python.org/3/
2. Документація Django. [Онлайн]. Доступно: https://docs.djangoproject.com/en/3.2/
3. Документація TextBlob. [Онлайн]. Доступно: https://textblob.readthedocs.io/en/dev/
4. HTML. Доступно: https://developer.mozilla.org/en-US/docs/Web/HTML
5. CSS:. [Онлайн]. Доступно: https://developer.mozilla.org/en-US/docs/Web/CSS
6. Deep Learning Book by Ian Goodfellow, Yoshua Bengio, and Aaron Courville [Онлайн]. Доступно: http://www.deeplearningbook.org/
7. Downey A. Think Python: How to Think Like a Computer Scientist, 2nd edition. Needham: Green Tea Press, 2015. 36 с
8. James A., Edward R. Talking Nets An Oral History of Neural Networks. Cambridge: The MIT Press, 2020. 10 с
9. Christian B. The Alignment Problem: Machine Learning and Human Values. New York: W. W. Norton & Company, 2020. 25 с.
10. Bradski G., Kaehler A. Learning OpenCV: Computer Vision with the OpenCV Library. Sebastopol: O'Reilly Media, Inc, 2011. 53 с.
11. Raschka S., Mirjalili V. Python Machine Learning. Livery Place: Packt Publishing Ltd, 2019. 296 с
12. Karat, C-M, Blom, J. O., & Karat, J. (2004). Designing Personalized User Experiences in eCommerce. Springer: Human–Computer Interaction Series. Springer, Netherlands, 348
13. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.
14. Chollet, F. (2018). Deep Learning with Python. Manning Publications.

15. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

16. Goldberg, Y. (2017). Neural Network Methods for Natural Language Processing. Morgan & Claypool Publishers.

17. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

18. Jurafsky, D., & Martin, J. H. (2021). Speech and Language Processing. Pearson.

19. Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980.

20. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature, 521(7553), 436-444.

21. Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

22. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.

23. Olah, C. (2015). Understanding LSTM Networks. Retrieved from URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

24. Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532-1543.

25. Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. arXiv preprint arXiv:1802.05365.

26. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving Language Understanding by Generative Pre-Training. OpenAI.

27. Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. Neural Networks, 61, 85-117.

28. Google AI Blog: BERT. (2018). Retrieved from URL: https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html Towards Data Science: Introduction to NLP. (2018). Retrieved from https://towardsdatascience.com/an-introduction-to-natural-language-processing-nlp-6a20621d0c93

29. arXiv: Machine Learning. Retrieved from URL: https://arxiv.org/archive/cs.LG

30. Zhang, Y., & Wallace, B. (2017). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 253-263.

31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems, 30.

ДОДАТКИ

urls.py

```
from django.urls import path
from django.contrib.auth import views as auth_views
from . import views
from .views import custom_login_view
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('', views.article_list, name='article_list'),
    path('trending/', views.trending, name='trending'),
    path('latest/', views.latest, name='latest'),
    path('listings/', views.listings, name='listings'),
    path('register/', views.register, name='register'),
    path('logout/', auth_views.LogoutView.as_view(), name='logout'),
    path('login/', custom_login_view, name='login'),
    path('news/<int:pk>/', views.article_detail, name='article_detail'),
    path('news/rate/<int:pk>/', views.rate_article, name='rate_article'),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

views.py

```
from django.shortcuts import render
from pyexpat import model
from .models import Article
from django.shortcuts import render, redirect
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth import login
from .forms import CustomUserCreationForm
from django.contrib.auth import login
from django.contrib.auth import authenticate, login
from django.contrib.auth.forms import AuthenticationForm
from django.shortcuts import render, get_object_or_404
import joblib
from django.shortcuts import render, get_object_or_404
from .models import Article, UserInteraction
from sklearn.feature_extraction.text import TfidfVectorizer
def article_list(request):
    articles = Article.objects.filter(link='general').order_by('-
created_at')
```

```
        return render(request, 'news/article_list.html', {'articles':
articles})
    def trending(request):
        trending_articles = Article.objects.filter(link='trending').order_by('-
created_at')
        return render(request, 'news/trending.html', {'articles':
trending_articles})
    def latest(request):
        latest_articles = Article.objects.filter(link='latest').order_by('-
created_at')
        return render(request, 'news/latest.html', {'articles':
latest_articles})
    def listings(request):
        listing_articles = Article.objects.filter(link='listings').order_by('-
created_at')
        return render(request, 'news/listings.html', {'articles':
listing_articles})
    def register(request):
        if request.method == 'POST':
            form = CustomUserCreationForm(request.POST)
            if form.is_valid():
                user = form.save()
                login(request, user)
                return render(request, 'news/register.html', {'form': form,
'registration_success': True})
        else:
            form = CustomUserCreationForm()
        return render(request, 'news/register.html', {'form': form})

    def custom_login_view(request):
        if request.method == 'POST':
            form = AuthenticationForm(request, data=request.POST)
            if form.is_valid():
                username = form.cleaned_data.get('username')
                password = form.cleaned_data.get('password')
                user = authenticate(request, username=username,
password=password)
                if user is not None:
                    login(request, user)
                    request.session['welcome_message'] = f"Приветствую,
{user.username}!"
                    return redirect('article_list')
```

```
            else:
                form = AuthenticationForm()
            return render(request, 'news/login.html', {'form': form})
        def article_detail(request, pk):
            article = get_object_or_404(Article, pk=pk)

            user = request.user
            if user.is_authenticated:
                recommended_articles = recommend_articles(user)
            else:
                recommended_articles = Article.objects.order_by('-created_at')[:5]

            context = {
                'article': article,
                'recommended_articles': recommended_articles,
            }

            return render(request, 'news/article_detail.html', context)

        def rate_article(request, pk):
            article = get_object_or_404(Article, pk=pk)

            if request.method == 'POST':
                rating = request.POST.get('rating')
                user = request.user

                interaction, created =
    UserInteraction.objects.get_or_create(user=user, article=article)

                interaction.rating = rating
                interaction.click_count += 1
                interaction.save()

                print(f"Saved rating for user {user.username}: {rating}, Article:
    {article.title}")

            return redirect('article_detail', pk=article.pk)


        def recommend_articles(user, num_recommendations=5):
            print(f"Рекомендація для користувача: {user.username} (ID: {user.id})")
```

```
        model_data = joblib.load('recommendation_model.pkl')
        model = model_data['model']
        category_encoder = model_data['category_encoder']
        tone_encoder = model_data['tone_encoder']
        user_encoder = model_data['user_encoder']

        try:
            user_encoded = user_encoder.transform([user.id])[0]
            print(f"Код пользователя: {user_encoded}")
        except ValueError:
            print(f"Пользователь {user.username} не знайдено в енкодері.")
            return Article.objects.order_by('-
created_at')[:num_recommendations]

        interacted_article_ids =
UserInteraction.objects.filter(user=user).values_list('article_id', flat=True)
        print(f"ID статей, з якими користувач взаємодіяв:
{list(interacted_article_ids)}")

        articles_to_recommend =
Article.objects.exclude(id__in=interacted_article_ids)
        print(f"Кількість статей для рекомендацій: {articles_to_recommend.count()}")

        # Підготовка даних
        data = []
        article_list = []
        for article in articles_to_recommend:
            try:
                features = [
                    user_encoded,
                    category_encoder.transform([article.category])[0],
                    tone_encoder.transform([article.tone])[0],
                ]
                data.append(features)
                article_list.append(article)
            except ValueError as e:
                print(f"Помилка {article.id}: {e}")
            continue

        if not data:
            print("Немає даних.")
```

```
            return Article.objects.order_by('-
created_at')[:num_recommendations]
        probabilities = model.predict_proba(data)[:, 1]  # Ймовірність 1
        print(f"Ймовірності: {probabilities}")

        articles_with_probs = list(zip(article_list, probabilities))
        articles_with_probs.sort(key=lambda x: x[1], reverse=True)
    print("Топ рекомендацій статей:")
        for article, prob in articles_with_probs[:num_recommendations]:
            print(f"Статья ID {article.id}, Заголовок: {article.title}, Ймовірність:
{prob:.4f}")
        recommended_articles = [article for article, prob in
articles_with_probs[:num_recommendations]]

        return recommended_articles
```

models.py

```
from django.db import models
from django.contrib.auth.models import User
from django.db import models
from django.contrib.auth.models import User

class Article(models.Model):
    CATEGORY_CHOICES = [
        ('fall', 'Падіння'),
        ('rise', 'Ріст'),
        ('listing', 'Листинг'),
        ('new', 'Нове'),
        ('promo', 'Промоакція'),
    ]

    LINK = [
        ('latest', 'Latest'),
        ('trending', 'Trending'),
        ('listings', 'Listings'),
        ('general', 'General'),
    ]

    TONE_CHOICES = [
        ('positive', 'Позитивний'),
        ('neutral', 'Нейтральний'),
```

```
            ('negative', 'Негативний'),
        ]

        title = models.CharField(max_length=200)
        content = models.TextField()
        short_description = models.TextField(blank=True, null=True)
        image = models.ImageField(upload_to='images/', blank=True, null=True)
        video = models.URLField(blank=True, null=True)
        created_at = models.DateTimeField(auto_now_add=True)
        updated_at = models.DateTimeField(auto_now=True)
        is_trending = models.BooleanField(default=False)
        category = models.CharField(max_length=10, choices=CATEGORY_CHOICES,
default='new')
        recommendations = models.ManyToManyField('self', blank=True,
related_name='recommended_articles')
        tone = models.CharField(max_length=10, choices=TONE_CHOICES,
default='neutral')
        link = models.CharField(max_length=10, choices=LINK, default='general')

        def __str__(self):
            return self.title

        def get_absolute_url(self):
            return f"/news/{self.id}/"

    class UserInteraction(models.Model):
        user = models.ForeignKey(User, on_delete=models.CASCADE)
        article = models.ForeignKey(Article, on_delete=models.CASCADE)
        rating = models.IntegerField(null=True, blank=True)
        viewed_at = models.DateTimeField(auto_now_add=True)
        click_count = models.IntegerField(default=0)

        def __str__(self):
            return f"{self.user.username} - {self.article.title}
({self.rating})"
```

admin.py

```
from django.contrib import admin
from .models import Article
from .models import UserInteraction
@admin.register(Article)
class ArticleAdmin(admin.ModelAdmin):
```

```
        list_display = ('title', 'created_at', 'category', 'is_trending',
'tone', 'link')
        list_filter = ('category', 'is_trending', 'tone', 'created_at')
        search_fields = ('title', 'content', 'short_description')
        ordering = ('-created_at',)


        fields = ('title', 'short_description', 'content', 'image', 'video',
'category', 'tone', 'is_trending', 'link')


        formfield_overrides = {
            'short_description': {'widget': admin.widgets.AdminTextareaWidget},
            'content': {'widget': admin.widgets.AdminTextareaWidget},
        }


admin.site.register(UserInteraction)
```