

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет**  
**імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

**ДОПУЩЕНО ДО ЗАХИСТУ**  
Завідувач кафедри інтелектуальних  
інформаційних систем, д-р техн. наук, проф.  
\_\_\_\_\_ Ю. П. Кондратенко  
«\_\_\_\_» \_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

**ІНТЕЛКТУАЛЬНА СИСТЕМА ОЦІНЮВАННЯ ЯКОСТІ**  
**МУЛЬТИМЕДІЙНОГО КОНТЕНТУ**

Спеціальність 122 «Комп'ютерні науки»

**122 – КРМ – 601.21817103**

*Виконав студент 6-го курсу, групи 601*  
\_\_\_\_\_ *Г.П.Клименко*  
«19» лютого 2024 р.

*Керівник: д.т.н., професор*  
\_\_\_\_\_ *О.П. Гожий*  
«19» лютого 2024 р.

**Миколаїв – 2024**

**Чорноморський національний університет ім. Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

Освітньо-кваліфікаційний рівень **магістр**

Галузь знань **12 «Інформаційні технології»**

*(шифр і назва)*

Спеціальність **122 «Комп'ютерні науки»**

*(шифр і назва)*

**ЗАТВЕРДЖУЮ**

Завідувач кафедри інтелектуальних  
інформаційних систем, д-р техн. наук, проф.

\_\_\_\_\_ Ю. П. Кондратенко

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**на виконання кваліфікаційної роботи**

**Клименко Григорію Павловичу**

*(прізвище, ім'я, по батькові)*

1. Тема кваліфікаційної роботи «Інтелектуальна система оцінювання якості мультимедійного контенту».

Керівник роботи Гожий Олександр Петрович д.т.н., професор.

Затв. наказом Ректора ЧНУ ім. Петра Могили від «19» жовтня 2023 р. № 201

2. Строк подання студентом роботи 19 лютого 2024 р.

3. Вхідні (початкові) дані до роботи: датасет фільмів, режисерів, сценаристів та рейтингу IMDb.

4. Зміст пояснювальної записки (перелік питань, які потрібно розглянути):

– дослідження сфери мультимедійного контенту;

– обґрунтування вибору інструментальних засобів розробки інтелектуальної системи для якості оцінювання мультимедійного контенту;

– розробка та здійснення програмної реалізації веб-додатку для оцінювання якості мультимедійного контенту;

5. Перелік графічного матеріалу: презентація, рисунки, таблиці.

6. Завдання до спеціальної частини: Охорона праці у комп'ютерному класі

7. Консультанти:

<b>Розділ</b>	<b>Прізвище, ініціали та посада консультанта</b>	<b>Підпис</b>
Спеціальна частина з охорони праці	д.б.н., Л.І. Григор'єва	
Методична частина	д.т.н., професор О.П. Гожий	

Керівник роботи д.т.н. професор, О.П.Гожий  
(наук. ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Завдання прийнято до виконання Клименко Г.П.  
(прізвище та ініціали)

\_\_\_\_\_ (підпис)

Дата видачі завдання « 31 » жовтня 2023 р.

# КАЛЕНДАРНИЙ ПЛАН

## виконання кваліфікаційної роботи магістра

Тема: Інтелектуальна система оцінювання якості мультимедійного контенту

№	Найменування роботи	Початок	Закінчення	Примітки
1	Визначення керівника і теми КРМ. Подання заяви на затвердження теми КРМ	01.09.2023	10.10.2023	Виконано
2	Отримання завдання на виконання КРМ	11.10.2023	01.11.2023	Виконано
3	Складання календарного плану на період виконання КРМ	02.11.2023	10.11.2023	Виконано
4	Огляд літератури за темою дослідження	11.11.2023	26.11.2023	Виконано
5	Проходження передатестаційної практики, збір та аналіз матеріалів до КРМ	27.11.2023	23.12.2023	Виконано
6	Аналіз предметної області та розробка технічного завдання. Моделювання результатів	25.12.2023	12.01.2024	Виконано
7	Опис фахової частини КРМ	13.01.2024	25.01.2024	Виконано
8	Розробка спеціальної частини з охорони праці та методичної частини	26.01.2024	02.02.2024	Виконано
9	Перший попередній захист КРМ на засіданні комісії кафедри	29.01.2024	29.01.2024	Виконано
10	Корегування роботи за результатами попереднього захисту	30.01.2024	05.02.2024	Виконано
11	Доробка та остаточне оформлення КРМ	06.02.2024	11.02.2024	Виконано
12	Другий попередній захист КРМ на засіданні комісії кафедри	12.02.2024	12.02.2024	Виконано
13	Подання КРМ, її електронної копії та інших документів (відгуку, рецензії) до захисту	19.02.2024	20.02.2024	Виконано
14	Захист КРМ перед екзаменаційною комісією (ЕК)	26.02.2024	27.02.2024	

Розробив студент Клименко Г.П. \_\_\_\_\_  
(прізвище та ініціали) (підпис)

Керівник роботи д.т.н., Гожий О.П. \_\_\_\_\_  
(наук. ступінь, вчене звання, прізвище та ініціали) (підпис)

«11» листопада 2023 р.

## **АНОТАЦІЯ**

до кваліфікаційної роботи магістра  
студента групи 601 ЧНУ ім. Петра Могили

**Клименка Григорія Павловича**

на тему: **“ІНТЕЛЕКТУАЛЬНА СИСТЕМА ОЦІНЮВАННЯ ЯКОСТІ  
МУЛЬТИМЕДІЙНОГО КОНТЕНТУ”**

**Об’єкт дослідження** – процес оцінювання якості мультимедійного контенту.

**Предмет дослідження** – інтелектуальні методи оцінювання якості мультимедійного контенту.

**Мета** – підвищення якості оцінювання мультимедійного контенту за рахунок використання методів нейронних мереж.

Пояснювальна записка індивідуального тому роботи складається зі вступу, 3 розділів, висновків, методичної частини та спеціальної частини з охорони праці.

Загальний обсяг роботи – 74 сторінок. Кваліфікаційна робота містить один додаток, 19 рисунків, 5 таблиць та посилання на 41 літературне джерело.

Ключові слова: Інтелектуальна система, мультимедійний контент, веб-додаток, оцінювання якості.

## **ABSTRACT**

to the master's qualification work by the student of the group 601 of Petro Mohyla  
Black Sea National University

**Klymenko Hryhorii Pavlovich**

### **“INTELLIGENT SYSTEM FOR QUALITY ASSESSMENT OF MULTIMEDIA CONTENT ”**

Object of research - the process of evaluating the quality of multimedia content.

Subject of research – the research is intellectual methods of evaluating the quality  
of multimedia content.

The purpose of the thesis – to improve the quality of multimedia content evaluation  
through the use of neural network methods.

Thesis consist of general part, and individual volumes.

The explanatory note of the individual volume of the work consists of an  
introduction, 3 sections, conclusions, a methodological part and a special part on labor  
protection.

The total volume of work is 69 pages. The qualification work contains one  
appendix, 19 figures, 5 tables and references to 41 literary sources.

Keywords: Intelligent system, multimedia content, web application, quality  
assessment.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1 ДОСЛІДЖЕННЯ СИСТЕМ АНАЛІЗУ МУЛЬТИМЕДІЙНОГО КОНТЕНТУ	6
1.1 Дослідження та аналіз сфери мультимедійного контенту та аналіз систем оцінювання якості .....	6
1.2 Оцінювання мультимедійного контенту .....	9
1.3 Технології оцінювання мультимедійного контенту .....	10
1.4 Огляд та аналіз сучасних технологій.....	12
Висновки до розділу 1 .....	13
2 ПЛАТФОРМИ ТА ІНСТРУМЕНТИ ДЛЯ СТВОРЕННЯ СИСТЕМИ ОЦІНЮВАННЯ.....	16
2.1 Редактор коду Visual Studio Code .....	16
2.2 Інструменти для створення аналітичної системи .....	19
Висновки до розділу 2 .....	27
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ОЦІНКИ КОНТЕНТУ .....	29
3.1 Архітектура аналітичної системи оцінювання .....	29
3.2 Використання технологій нейронних мереж в системах оцінювання контенту.....	32
3.3 Розробка веб-додатку .....	41
3.4 Мобільна адаптація веб-додатку .....	46
Висновки до розділу 3 .....	54
ВИСНОВКИ .....	55
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	57
ДОДАТОК А Код програми .....	61

## **ПЕРЕЛІК СКОРОЧЕНЬ**

AJAX - Asynchronous JavaScript and XML  
API - Application Programming Interface  
AR - Augmented Reality  
CSS - Cascading Style Sheets  
FTP - File Transfer Protocol  
GPU - Graphics Processing Unit  
HD - High Definition  
HTML - HyperText Markup Language  
JSON - JavaScript Object Notation  
KB - Kilobyte  
MVC - Model-View-Controller  
NumPy - Numerical Python  
ORM - Object-Relational Mapping  
PCA - Principal Component Analysis  
ReLU - Rectified Linear Unit  
ROI - Return On Investment  
SEO - Search Engine Optimization  
t-SNE - t-Distributed Stochastic Neighbor Embedding  
SQL - Structured Query Language  
URL - Uniform Resource Locator  
VR - Virtual Reality  
VS Code - Visual Studio Code  
БШП - Багатошаровий перцептрон



## ВСТУП

У сучасному цифровому світі ми стаємо свідками небувалого зростання обсягів мультимедійного контенту, що створюється, поширюється та споживається щодня. Розвиток цифрових технологій, зокрема вдосконалення алгоритмів стиснення даних, збільшення пропускної спроможності мереж і поява передових мобільних пристроїв, сприяли масовому розповсюдженню відео, аудіо, графічних матеріалів та інтерактивного контенту. Ця тенденція призвела до значного зростання вимог до якості та релевантності мультимедійного контенту з боку споживачів, які прагнуть отримати максимально збагачений та задовільний досвід від його використання.

Традиційні методи оцінювання якості мультимедійного контенту часто обмежуються суб'єктивними оцінками користувачів або зосереджуються на базових технічних параметрах, таких як роздільна здатність зображення, бітрейт аудіо, частота кадрів відео та інші. Втім, такі підходи рідко враховують глибинний вплив контенту на емоційне та когнітивне сприйняття аудиторії, що є ключовим для забезпечення високої якості досвіду споживання. Так, незважаючи на технічну досконалість, контент може не викликати очікуваного резонансу у споживача через недоліки у структурі, контексті або змісті.

У цьому контексті, пошук шляхів автоматизованого аналізу та оцінювання мультимедійного контенту, які б враховували широкий спектр параметрів — від технічних характеристик до емоційної відповіді аудиторії, стає нагальною потребою. Це зумовлює необхідність розробки інтелектуальних систем, здатних комплексно оцінювати якість мультимедійного контенту, використовуючи досягнення в областях машинного навчання, штучного інтелекту, обробки природної мови та комп'ютерного зору.

Ця дипломна робота присвячена розробці інтелектуальної системи оцінювання якості мультимедійного контенту.

**Мета роботи:** Метою дипломної роботи є розробка інтелектуальної системи, заснованої на нейронних мережах для оцінювання мультимедійного контенту (для

оцінювання рейтингу фільму). Система використовує набір даних, що включає інформацію про режисерів, сценаристів, жанри фільмів та інші релевантні параметри. Основна задача програми полягає в тому, щоб на основі цих даних оцінити рейтинг фільму, здійснити розподіл аудиторії за статтю, віком та професією, забезпечивши таким чином цінну інформацію для кінопродюсерів, маркетингологів, розповсюджувачів фільмів, а також для широкого кола глядачів.

**Об'єкт дослідження:** Процес оцінювання мультимедійного контенту

**Предмет дослідження:** Інтелектуальні методи аналізу і оцінювання мультимедійного контенту.

# **1 ДОСЛІДЖЕННЯ СИСТЕМ АНАЛІЗУ МУЛЬТИМЕДІЙНОГО КОНТЕНТУ**

## **1.1 Дослідження та аналіз сфери мультимедійного контенту та аналіз систем оцінювання якості**

Якість мультимедійного контенту є одним з визначальних елементів цифрової епохи, яка сформувалася на кордоні століть і продовжує еволюціонувати з кожним днем. Від відео, що транслюються на гігантських екранах, до ілюстрацій у соціальних мережах, якість контенту впливає на все: від користувацького досвіду до економічної віддачі. Якість мультимедійного контенту охоплює широкий спектр характеристик — від чистоти звуку та чіткості зображення до глибини сюжету та здатності викликати емоційну реакцію.

У сфері цифрового маркетингу та реклами, де боротьба за увагу користувача є нещадною, якість контенту може бути вирішальним фактором, що визначає успіх чи невдачу кампанії. Цифровий контент стикається з викликом вижити в умовах інформаційного шуму, де користувачі щодня бомбардуються тисячами повідомлень. Тому мультимедійний контент повинен не лише привертати увагу, але й бути достатньо якісним, щоб утримати користувача і спонукати його до дії.

З погляду контент-креаторів, якість контенту є відображенням їх ремесла. У світі, де контент є валютою, якість стає маркером професіоналізму та оригінальності. Для кінематографістів, музикантів, художників та письменників, можливість створювати високоякісний контент означає здатність виразити свою творчу візію без компромісів, що забезпечує глибше занурення в твір і створення більш багатогранного досвіду для аудиторії.

Технічна якість мультимедійного контенту, також має безпосередній вплив на користувацьке сприйняття. У світі, де HD, 4K, та навіть 8K стають стандартами, очікування користувачів до якості зображення та звуку неухильно зростають. Це

вимагає від технологій постійного вдосконалення та від креаторів — знань та навичок для роботи з новітнім обладнанням [1].

З іншого боку, володіння технологіями не завжди гарантує створення високоякісного контенту. Ключову роль відіграє змістовна якість: наскільки глибокими, інформативними, емоційними чи переконливими є текстові, візуальні, або аудіальні компоненти. Якість сюжету, здатність до розповіді історій, рівень креативності та інноваційність — усе це стає необхідним для виділення контенту серед маси однотипних повідомлень .

Інтелектуальні системи оцінювання якості мультимедійного контенту, які використовують штучний інтелект і аналітику великих даних, можуть стати рішенням для об'єктивізації та автоматизації цього процесу. Зі зростаючою кількістю контенту та його різноманітністю, такі системи надзвичайно важливі для ринку мультимедіа. Вони дозволяють не тільки оцінювати існуючий контент, але й дають змогу креаторам зрозуміти, які аспекти їхньої роботи сприймаються позитивно, а які потребують вдосконалення. Такий підхід сприяє підвищенню загального рівня якості мультимедійного контенту, що є вигідним для всіх учасників медіа-простору.

Інтерес до якості мультимедійного контенту постає як ключовий елемент у стратегічних планах різних стейкхолдерів індустрії, оскільки він безпосередньо впливає на залучення та утримання аудиторії, а також на генерування прибутку. Якість, у цьому контексті, визначається не тільки як технічна досконалість, але й як відповідність змісту очікуванням та вимогам споживачів. Вона включає в себе глибину та оригінальність контенту, відповідність його цільовій аудиторії та здатність створювати значущі враження.

Контент-креатори зацікавлені у високій якості своїх продуктів, оскільки це безпосередньо впливає на їхнє професійне зростання та репутацію. Вони прагнуть створювати контент, що резонує з аудиторією, викликає емоційний відгук та заохочує поділитися ним з іншими. Це забезпечує вищу залученість та розширення охоплення, що, в свою чергу, підвищує шанси на монетизацію через рекламу, продажі, ліцензування контенту чи залучення спонсорських коштів [2].

З точки зору маркетингових відділів та рекламодавців, якість контенту є фундаментом ефективної реклами. Високоякісний контент приваблює більше переглядів та поділень, що збільшує рекламні можливості та веде до кращого ROI [3]. Рекламодавці охочіше інвестують у платформи та канали, які демонструють високу якість контенту, оскільки це зменшує ризик асоціації їхніх брендів із низькоякісним чи неприйнятним контентом.

Для платформ, які розповсюджують мультимедійний контент, таких як стрімінгові сервіси, соціальні мережі та відеохостинги, забезпечення високоякісного контенту є ключем до утримання користувачів. Це також допомагає у відборі та просуванні контенту, який максимально відповідає інтересам та перевагам їхньої аудиторії, забезпечуючи персоналізований досвід.

Інвестори та акціонери теж стежать за якістю контенту, оскільки вона впливає на стабільність та зростання компаній, у які вони вкладають кошти. Якісний контент сприяє позитивній репутації бренду, що веде до збільшення вартості акцій та залучення нових інвестицій [4].

Таким чином, інтерес з боку стейкхолдерів до оцінювання якості контенту має фундаментальне значення для всієї мультимедійної індустрії. Якісний контент є двигуном інновацій, забезпечує економічний успіх та підтримує репутаційний капітал, створюючи цінність для всіх учасників від творців до кінцевих користувачів.

Якість контенту є фундаментальним показником користувацького досвіду в сучасному медіа-просторі. Це важливо не тільки через естетичне задоволення, яке користувачі отримують від високої роздільної здатності зображення чи чистоти звуку, але й через глибшу взаємодію з контентом, яка вимірюється залученістю та емоційною реакцією.

У сфері цифрового контенту, де користувачі мають необмежений доступ до різноманітного медіа, вибір часто визначається якістю. Ця якість визначається не тільки технічними параметрами, але й змістовністю, креативністю, оригінальністю та релевантністю контенту. Це може включати динаміку розповіді, візуальну

привабливість, інтерактивність та інші аспекти, які забезпечують занурення користувачів у досвід, що надається мультимедійним контентом [5].

Задоволення користувачів є безпосередньо пов'язаним із якістю контенту, оскільки високоякісний контент може підвищувати емоційний тон, надихати, інформувати, або навчати. Він може створювати враження новизни та інноваційності, що є особливо важливим в контексті постійної конкуренції за увагу користувача. Якість контенту також сприяє вибудовуванню довіри користувачів до брендів та платформ, оскільки вона відображає їхню прихильність до високих стандартів та повагу до аудиторії.

Залученість користувачів є іншим критичним елементом, який визначається якістю контенту. Високоякісний контент заохочує користувачів не просто споживати інформацію, але й активно взаємодіяти з нею — через коментарі, поділи, і створення спільноти навколо контенту [6]. Це не тільки покращує видимість контенту в соціальних мережах та інших платформах, але й створює цінність для користувачів, оскільки вони відчують себе частиною більшого діалогу чи спільноти.

В кінцевому рахунку, роль якості контенту в користувацькому досвіді не може бути недооцінена. Вона служить як каталізатор для залучення та утримання аудиторії, стимулює взаємодію та обговорення, і підтримує створення глибоких та тривалих відносин між користувачами та творцями контенту.

## **1.2 Оцінювання мультимедійного контенту**

Оцінювання мультимедійного контенту є критичним процесом, який має далекосяжні наслідки для всіх учасників медіа-екосистеми. Кожна група стейкхолдерів — від кінцевих користувачів та контент-мейкерів до платформ, які розповсюджують контент — отримує унікальні переваги від високоякісного мультимедійного контенту, який був відповідно оцінений та класифікований.

Для користувачів, якість контенту безпосередньо впливає на їхнє задоволення та досвід. Високоякісний контент забезпечує глибше занурення у перегляд, слухання чи взаємодію з контентом, що підвищує емоційне зв'язування

та лояльність до брендів або платформ. Це також може сприяти покращенню освітнього та інформаційного потенціалу контенту, що особливо важливо у сфері навчання та розвитку.

Контент-мейкери, зі свого боку, використовують оцінювання контенту як засіб самовдосконалення та маркетингу. Точне розуміння того, що високо цінується серед їх аудиторії, допомагає їм адаптувати та вдосконалювати свої творчі підходи, що, у свою чергу, може підвищити їх видимість, популярність та доходи. Крім того, високоякісний контент часто асоціюється з професіоналізмом та експертизою, що може підсилити їх репутацію та авторитет в галузі [7].

Платформи, які розповсюджують мультимедійний контент, також отримують вигоду з оцінювання якості. Це дає їм змогу керувати контентом, пропонуючи користувачам найбільш релевантний та високоякісний контент, що може підвищити час перебування на платформі та збільшити частоту відвідувань. Оцінювання якості також допомагає в оптимізації алгоритмів рекомендацій, що є ключовим компонентом утримання аудиторії і залучення нових користувачів. З точки зору монетизації, платформи можуть використовувати дані про якість контенту для розробки преміальних пропозицій, що знову ж таки підвищує прибутки.

В цілому, систематичне оцінювання мультимедійного контенту сприяє створенню екосистеми, де якість є ключовим драйвером для всіх стейкхолдерів. Це стимулює інновації, підвищує стандарти виробництва контенту та покращує загальний медіа-продукт, яким користуються мільйони людей щодня. Інтелектуальні системи оцінювання, які можуть обробляти великі обсяги даних та надавати точні метрики, є необхідними для підтримки та розвитку цієї екосистеми, забезпечуючи, щоб кожен контент, який досягає користувачів, був найвищої якості [8].

### **1.3 Технології оцінювання мультимедійного контенту**

Технології оцінювання мультимедійного контенту розвивалися паралельно із стрімким зростанням цифрових медіа, відповідаючи на зростаючу потребу в аналізі

та класифікації контенту на основі його якості. Ця область об'єднує різноманітні дисципліни, включаючи обробку сигналів, машинне навчання, комп'ютерний зір, та обробку природної мови, і спрямована на створення систем, здатних ефективно та об'єктивно оцінювати мультимедійний контент [9].

Сучасні технології оцінювання виходять за рамки простих параметрів, таких як роздільна здатність зображення чи бітрейт аудіо. Вони включають складні алгоритми, які можуть аналізувати все — від якості кольоропередачі та світлових умов у відео до семантичного змісту та емоційного тону в аудіозаписах. Використання машинного навчання та штучного інтелекту дозволяє розробляти моделі, які можуть навчатися від реальних даних і ставати все кращими в ідентифікації тонких нюансів, що впливають на сприйняття якості людьми.

Ключові технології включають алгоритми розпізнавання образів, які можуть визначати композиційні характеристики, такі як правило третин чи баланс світла та тіні в зображеннях, а також системи рекомендацій, які використовують великі набори даних для передбачення інтересів користувачів і пропонування контенту, який їх найбільш ймовірно зацікавить.

Зв'язок між технологіями оцінювання та їх застосуванням у контексті мультимедіа є взаємозалежним [10]. З одного боку, технології повинні бути достатньо розвинені, щоб адекватно оцінювати контент. З іншого — вони мають адаптуватися до змін у споживацьких перевагах і трендах мультимедіа. Наприклад, з появою віртуальної реальності та 360-градусних відео, з'явилася потреба в нових методах оцінювання, які можуть враховувати унікальні вимоги до цих форматів.

Технології оцінювання також відіграють важливу роль у процесі виробництва мультимедійного контенту, дозволяючи контент-мейкерам отримувати зворотній зв'язок про якість їхніх творів у режимі реального часу та робити корективи до завершення проекту. Це сприяє створенню більш якісного продукту та може значно скоротити час та витрати на постпродакшн.

В цілому, технології оцінювання мультимедійного контенту стають дедалі більш витонченими та інтегрованими у всі аспекти виробництва та споживання



медіа, перетворюючи якість з абстрактного поняття на кількісно вимірюваний, аналізований та оптимізований параметр.

#### **1.4 Огляд та аналіз сучасних технологій**

У сучасному світі технології для оцінки контенту є необхідністю в різноманітних сферах, включаючи цифровий маркетинг, освіту, медіа та розваги. Вони дозволяють аналізувати, оцінювати та керувати контентом завдяки застосуванню машинного навчання, штучного інтелекту, аналізу даних, обробці природної мови та інших передових технологій.

Машинне навчання та штучний інтелект революціонізували підхід до модерації контенту та персоналізації, дозволяючи автоматично фільтрувати небажаний контент і створювати індивідуалізовані рекомендації для користувачів. Це сприяє підвищенню залученості та задоволення від використання платформ.

Обробка природної мови відкриває можливості для глибокого аналізу текстового контенту, включаючи аналіз емоційного забарвлення та автоматичне розподілення матеріалів за темами. Це полегшує управління великими обсягами інформації та оптимізацію контенту під потреби аудиторії.

Технології аналітики даних, такі як веб-аналітика, забезпечують детальне вивчення поведінки користувачів та ефективності контенту, допомагаючи маркетологам оптимізувати стратегії взаємодії [11].

Блокчейн пропонує новітні рішення для захисту авторських прав та забезпечення прозорості використання цифрового контенту, гарантуючи авторам визнання їхньої праці та справедливу винагороду.

Штучний інтелект також відіграє ключову роль у розпізнаванні образів та відеоаналітиці, автоматизуючи процес класифікації візуального контенту та його модерацію.

Зрештою, сучасні платформи для створення інтерактивного контенту дозволяють розробляти захоплюючі інтерактивні досвіди, що підвищують залученість користувачів та розширюють можливості для творчості.

Ці технології сприяють створенню більш персоналізованого, інтерактивного та безпечного цифрового простору, відкриваючи нові можливості для розвитку та інновацій у сфері цифрового контенту.

У сфері цифрового маркетингу, технології аналізу контенту відіграють вирішальну роль, дозволяючи маркетологам оптимізувати контент для підвищення залученості та ефективності. Основні напрямки включають:

- Персоналізація і SEO: Машинне навчання та ШІ сприяють створенню персоналізованих маркетингових кампаній, аналізуючи поведінку користувачів та оптимізуючи контент під пошукові системи. Це забезпечує високу релевантність контенту для конкретної аудиторії та покращує його видимість [12].
- Аналітика та вимірювання: Використання інструментів веб-аналітики допомагає визначити, які типи контенту найбільш ефективні, дозволяючи маркетологам зосередитися на кращих каналах розповсюдження та оптимальних часових рамках для публікацій [13].
- Відеоконтент та інтерактивність: Розвиток технологій відеоаналітики та інтерактивних форматів, таких як AR та VR, відкриває нові можливості для залучення аудиторії. Відеоконтент залишається одним із найбільш ефективних засобів взаємодії, а додавання інтерактивних елементів може значно підвищити зацікавленість та взаємодію з брендом [15].

Загалом, інтеграція цих технологій у стратегії цифрового маркетингу дозволяє створювати більш персоналізовані, залучаючі та вимірювані кампанії, підвищуючи їхню загальну ефективність та вплив на поведінку користувачів.

## **Висновки до розділу 1**

У рамках дослідження систем аналізу мультимедійного контенту було проведено вивчення сфери мультимедійного контенту, а також аналіз існуючих систем оцінювання якості. Було розглянуто різноманітні підходи до оцінювання мультимедійного контенту, включаючи наявні технології та методики, а також провели огляд та аналіз сучасних технологій та аналогів у цій галузі.

Дослідження показало, що сфера мультимедійного контенту динамічно розвивається, а потреби користувачів у якісному контенті зростають. Це створює високий попит на ефективні системи оцінювання якості, які можуть надавати об'єктивні та точні оцінки. Сучасні технології, зокрема нейромережі, відкривають нові можливості для розробки таких систем. Вони дозволяють не лише аналізувати технічні параметри контенту, але й враховувати емоційне сприйняття та вподобання користувачів.

Незважаючи на значний прогрес у цій галузі, існують виклики, пов'язані з об'єктивністю оцінок та адаптацією систем до швидко змінюваних трендів мультимедійного контенту. Також важливим аспектом є забезпечення конфіденційності та безпеки даних користувачів.

Для реалізації проекту з розробки інтелектуальної системи оцінювання якості мультимедійного контенту за допомогою нейромережі та інтеграції цієї системи у веб-застосунків, необхідно виконати наступні задачі:

1. Провести детальний аналіз зібраних даних на предмет виявлення викидів та пропусків, що можуть вплинути на якість тренування моделі.
2. У разі виявлення відхилень, застосувати методи та підходи для очищення даних, включаючи видалення або корекцію викидів та заповнення пропусків, для забезпечення високої якості вхідних даних для тренування моделі.
3. Розробити архітектуру нейромережі, оптимізовану для аналізу мультимедійного контенту, з врахуванням особливостей даних та поставлених цілей проекту.
4. Тренувати нейромережу на підготовлених даних, використовуючи визначені параметри та техніки оптимізації для досягнення високої точності прогнозування.
5. Провести оцінку точності та ефективності нейромережі, використовуючи тестові набори даних та методи валідації, для забезпечення високої якості рекомендацій.

6. Створити веб-додаток з десктопною та мобільною версіями, який забезпечить зручний інтерфейс для взаємодії користувачів з системою оцінювання мультимедійного контенту.
7. Інтегрувати розроблену нейромережу у веб-застосунок, забезпечивши ефективну взаємодію між користувачами, веб-інтерфейсом та системою оцінювання.

Виконання цих задач дозволить створити ефективну та надійну систему для оцінювання якості мультимедійного контенту, яка зможе задовольнити потреби користувачів у швидкому та точному аналізі контенту

## 2 ПЛАТФОРМИ ТА ІНСТРУМЕНТИ ДЛЯ СТВОРЕННЯ СИСТЕМИ ОЦІНЮВАННЯ

### 2.1 Редактор коду Visual Studio Code

Visual Studio Code, який також зазвичай називають VS Code — це редактор початкового коду, створений Microsoft із Electron Framework для Windows, Linux і macOS. Функції включають підтримку налагодження, підсвічування синтаксису, інтелектуальне завершення коду, фрагменти, рефакторинг коду та вбудований Git. Користувачі можуть змінювати тему, комбінації клавіш, параметри та встановлювати розширення, які додають функціональність [15].

В опитуванні розробників Stack Overflow 2022 серед 71 010 респондентів Visual Studio Code назвали найпопулярнішим інструментом середовища розробника, при цьому 74,48% повідомили [16], що вони ним користуються (див.рис.2.1).

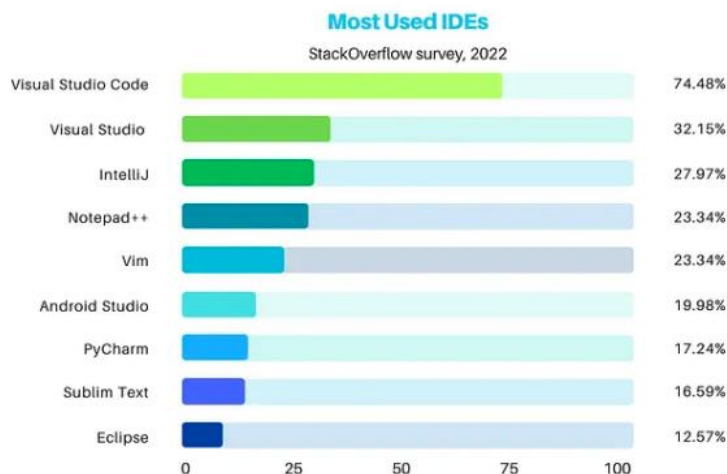


Рисунок 2.1 – Графік найпопулярніших середовищ розробки

Цей редактор заснований на платформі Electron, яка є популярним вибором для розробки веб-додатків, орієнтованих на Node.js, використовуючи механізм рендеринга Blink. Однією з ключових особливостей VS Code є використання компонента редактора "Monaco", який також використовується в Azure DevOps (колишній Visual Studio Online та Visual Studio Team Services).

## **Основні Можливості та Розширення**

VS Code постачається з базовою підтримкою більшості поширених мов програмування. Це означає, що користувачі мають доступ до таких функцій, як підсвічування синтаксису, зіставлення дужок, згортання коду та персоналізовані фрагменти коду. Додатково, є вбудована підтримка IntelliSense для мов, як JavaScript, TypeScript, JSON, CSS, HTML, а також можливості налагодження Node.js. Розширення для додаткових мов і функцій доступні через VS Code Marketplace, що робить його надзвичайно адаптивним і гнучким [17].

## **Робочі Області та Мовна Агностика**

Відмінною рисою VS Code є відсутність традиційної системи проектів, що дозволяє користувачам відкривати один або кілька каталогів одночасно. Ці каталоги можна зберегти у вигляді робочих областей для подальшого використання. Ця функція робить VS Code мовно-незалежним редактором, що ідеально підходить для роботи з різними мовами програмування. Користувачі можуть налаштовувати відображення файлів і папок у проекті, виключаючи непотрібні елементи [18].

## **Розширення та Персоналізація**

Однією з найважливіших функцій VS Code є його розширюваність. Користувачі можуть завантажувати та встановлювати розширення з центрального сховища, які додають нові можливості редактора та підтримку мов. Це включає в себе створення розширень для нових мов, тем, налагоджувачів, статичного аналізу коду, лінтерів, використовуючи Language Server Protocol [19].

## **Вбудоване Управління Версіями**

Управління версіями є вбудованою функцією VS Code. Вона включає спеціальну вкладку для доступу до налаштувань керування версіями та перегляду змін у поточному проекті. Підтримуються різні системи керування версіями, такі як Git, Apache Subversion, Perforce, дозволяючи користувачам створювати репозиторії та виконувати основні операції безпосередньо з редактора [20].

## **FTP Розширення та Веб-Розробка**

Завдяки розширенням FTP, VS Code можна використовувати як повнофункціональне рішення для веб-розробки, здійснюючи синхронізацію коду між локальним редактором і сервером [22].

### **Адаптивність та Крос-Платформенність**

VS Code дозволяє налаштувати кодову сторінку, символ нового рядка, та мову програмування для активного документа, роблячи його ідеальним інструментом для різноманітних платформ, локальних мереж та мов програмування.

### **Телеметрія та Прозорість**

Хоча Visual Studio Code збирає дані про використання для поліпшення продукту, ця функція може бути вимкнена користувачем. Важливо відзначити, що весь код телеметрії є відкритим, що забезпечує прозорість та дозволяє спільноті бачити, які саме дані збираються. Це демонструє відкритість Microsoft до зворотного зв'язку та співпраці зі спільнотою розробників [23].

Використання Visual Studio Code, його широкий спектр функцій, гнучкість та адаптивність до різних мов і технологій роблять його одним із найкращих виборів для розробників. Його розширюваність, інтеграція з системами керування версіями та підтримка різноманітних мов програмування забезпечують високу продуктивність та ефективність у робочому процесі.

Також важливо зазначити, що VS Code має велику кількість гарячих клавіш [24], що значно полегшують роботу з кодом. VS Code має можливість переназначати їх (див.рис.2.2).

Кафедра інтелектуальних інформаційних систем  
Інтелектуальна система оцінювання якості мультимедійного контенту

Visual Studio Code		Search and replace			
Keyboard shortcuts for macOS		⌘F	Find	⇧⌘T	Reopen closed editor
		⇧⌘F	Replace	⇧⌘Enter	Keep preview mode editor open
		⌘G / ⇧⌘G	Find next/previous	⌘Tab / ⇧⌘Tab	Open next / previous
		⇧⌘E	Select all occurrences of Find match	⇧⌘P	Copy path of active file
		⇧⌘D	Add selection to next Find match	⇧⌘R	Reveal active file in Finder
		⇧⌘D	Move last selection to next Find match	⇧⌘O	Show active file in new window/instance
		<b>Rich languages editing</b>		<b>Display</b>	
		⌘Space, ⇧E	Trigger suggestion	⌘F1	Toggle full screen
		⌘Space	Trigger parameter hints	⇧⌘E	Toggle editor layout (horizontal/vertical)
		⇧⌘F	Format document	⇧⌘= / ⇧⌘-	Zoom in/out
		⇧⌘F	Format selection	⇧⌘B	Toggle Sidebar visibility
		F12	Go to Definition	⇧⌘E	Show Explorer / Toggle focus
		⇧⌘F12	Peek Definition	⇧⌘F	Show Search
		⇧⌘F12	Open Definition to the side	⇧⌘G	Show Source Control
		⇧⌘F	Show References	⇧⌘D	Show Debug
		F2	Rename Symbol	⇧⌘X	Show Extensions
		⇧⌘X	Trim trailing whitespace	⇧⌘R	Replace in files
		⇧⌘M	Change file language	⇧⌘J	Toggle Search details
		<b>Navigation</b>		⇧⌘U	Show Output panel
		⇧⌘T	Show all Symbols	⇧⌘V	Open Markdown preview to the side
		⌘G	Go to Line...	⇧⌘Z	Zen Mode (Esc Esc to exit)
		⇧⌘P	Go to File...	<b>Debug</b>	
		⇧⌘O	Go to Symbol...	F9	Toggle breakpoint
		⇧⌘M	Show Problems panel	F5	Start/Continue
		F8 / ⇧F8	Go to next/previous error or warning	F11 / ⇧F11	Step into/ out
		⇧⌘Tab	Navigate editor group history	F10	Step over
		⇧⌘←	Go back/forward	⇧F5	Stop
		⇧⌘M	Toggle Tab moves focus	⇧⌘I	Show hover
		<b>Editor management</b>		<b>Integrated terminal</b>	
		⇧⌘W	Close editor	⇧⌘`	Show integrated terminal
		⇧⌘F	Close folder	⌘`	Create new terminal
		⇧⌘E	Split editor	⇧⌘C	Copy selection
		⇧⌘1 / ⇧⌘2 / ⇧⌘3	Focus into 1 <sup>st</sup> , 2 <sup>nd</sup> , 3 <sup>rd</sup> editor group	⇧⌘T / ⇧⌘L	Scroll up/down
		⇧⌘⇧ / ⇧⌘⇨ / ⇧⌘⇩ / ⇧⌘⇪	Move editor left/right	⇧⌘Up / ⇧⌘Dn	Scroll page up/down
		⇧⌘⇧	Move active editor group	⇧⌘Home / ⇧⌘End	Scroll to top/bottom
		<b>File management</b>			
		⇧⌘N	New File		
		⇧⌘O	Open File...		
		⇧⌘S	Save		
		⇧⌘S	Save As...		
		⇧⌘S	Save All		
		⇧⌘W	Close		
		⇧⌘⇧W	Close All		

Рисунок 2.2 – Гарячі клавіші VSCode

## 2.2 Інструменти для створення аналітичної системи

Python - це високорівнева, об'єктно-орієнтована мова програмування, яка вирізняється своїм простим і зрозумілим синтаксисом. Основа Python - синтаксис англійської мови, що робить його однією з найпростіших мов програмування для вивчення. Python дозволяє користувачам виконувати складні маніпуляції з даними, а також числовий аналіз, використовуючи структури даних, такі як датафрейми [25]. Ця багатоцільова мова програмування застосовна майже в будь-якій ситуації, де використовуються дані, лінії коду або математичні обчислення.

Python - найшвидше розвивається мова програмування, що використовується сьогодні. Він може бути використаний як для невеликих завдань, таких як управління ботом-модератором на Reddit, так і для більш складних зусиль, наприклад, для роботи з величезними обсягами фінансових даних хедж-фондів [26]. Оскільки це програмне забезпечення є вільним та відкритим, воно має величезну спільноту користувачів по всьому світу.

Python має багато професійних застосувань у світі великих даних і ряд бібліотек, які є корисними для тих, хто працює в області аналізу даних.



Розширюючи це опис, можна зазначити, що Python відомий своєю відносною простотою та високим рівнем читабельності коду. Ця особливість значно спрощує процес навчання та роботи з мовою, роблячи її доступною навіть для початківців у програмуванні. Додатково, з огляду на її об'єктно-орієнтований характер, Python дозволяє розробникам створювати модульні та масштабовані програми [27], що є ключовим для багатьох сучасних програмних рішень.

У сфері обробки даних Python надзвичайно потужний, завдяки своїм розширеним бібліотекам, таким як Pandas для обробки датафреймів, NumPy для складних математичних обчислень та SciPy для наукових та технічних застосувань. Крім того, Python є основою для багатьох бібліотек машинного навчання та штучного інтелекту, включаючи TensorFlow та Keras, роблячи його

### 2.2.1 Бібліотека NumPy

NumPy, що є скороченням від "Numerical Python", є одним з ключових пакетів Python для наукових обчислень. Ця бібліотека складається з багатовимірних масивних об'єктів, а також набору рутин, призначених для їх обробки [28]. NumPy є потужним інструментом для виконання різноманітних логічних та математичних завдань (див. рис.2.3).

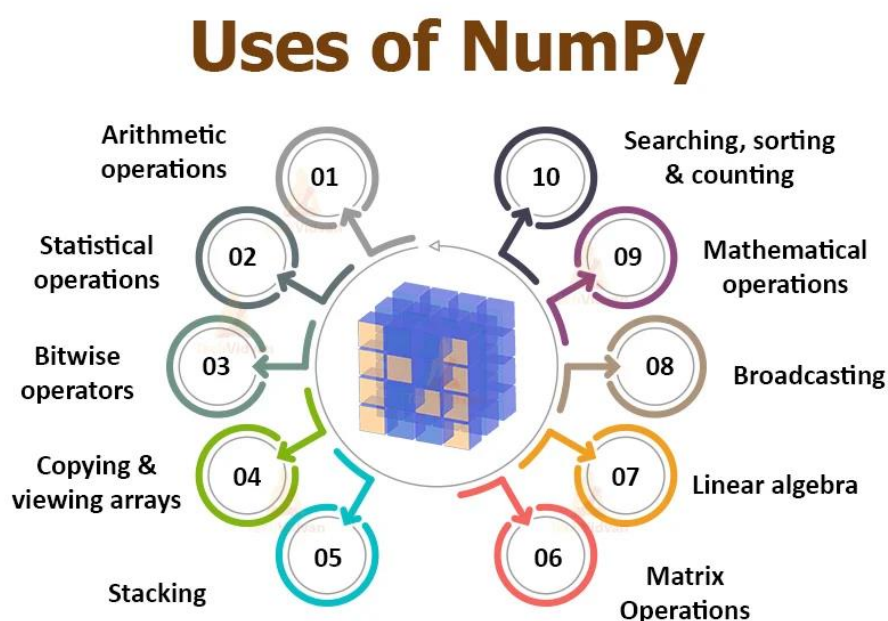


Рисунок 2.3 – Використання NumPy

Ось деякі з основних переваг використання NumPy для аналізу даних:

1. Багатовимірність даних: NumPy особливо корисний для створення об'єктів даних з  $N$  вимірами [29]. Це дозволяє користувачам ефективно працювати зі складними наборами даних, які мають багато вимірів, наприклад, у сферах фізики, інженерії чи фінансів.

2. Швидкість та ефективність: Фреймворк NumPy працює швидко та плавно при роботі з однорідними наборами даних [30]. Він оптимізований для обробки великих обсягів даних, забезпечуючи більш високу продуктивність, ніж традиційні підходи Python.

3. Ефективне використання пам'яті: При виконанні числових розрахунків масиви NumPy використовують менше пам'яті, ніж списки Python [31]. Також NumPy дозволяє користувачам вказувати типи даних у вмісті, що може оптимізувати код.

4. Ефективне зберігання та обробка даних: NumPy може ефективно зберігати дані та операції з даними, особливо коли масиви збільшуються за розміром. Це робить його ідеальним вибором для роботи з великими наборами даних [32].

5. Спрощення математичних операцій: Виконання математичних операцій на даних, збережених у NumPy, не є складним. NumPy надає широкий спектр функцій та інструментів для обробки та аналізу даних [33].

6. Підвищення швидкості робочого процесу: NumPy дозволяє користувачам збільшити швидкість своїх робочих процесів, забезпечуючи швидке обчислення та аналіз даних.

7. Інтеграція з іншими пакетами Python: NumPy може взаємодіяти з іншими пакетами Python. Оскільки NumPy існує вже досить довгий час, майже всі пакети машинного навчання та аналізу даних для Python використовують NumPy у якійсь мірі.

Ці особливості роблять NumPy незамінним інструментом у світі аналізу даних, наукових обчислень та розробки алгоритмів машинного навчання.

### **2.2.2 Бібліотека TensorFlow**

TensorFlow — це безкоштовна бібліотека програмного забезпечення з відкритим кодом для машинного навчання та штучного інтелекту . Його можна використовувати в ряді завдань, але він зосереджений на навчанні та висновках глибоких нейронних мереж .

TensorFlow розроблено командою Google Brain для внутрішнього використання Google у дослідженнях і виробництві. Початкову версію було випущено під ліцензією Apache 2.0 у 2015 році. Google випустив оновлену версію TensorFlow під назвою TensorFlow 2.0 у вересні 2019 року [34].

TensorFlow можна використовувати в багатьох мовах програмування, включаючи Python , JavaScript , C++ і Java . Ця гнучкість дає змогу використовувати різноманітні програми в різних секторах.

TensorFlow забезпечує можливість створення діаграм потоків даних, які ілюструють переміщення даних через створений графік. У такому графіку вузли символізують математичні операції, тоді як ребра або зв'язки між ними представляють собою багатовимірні масиви даних. Ці масиви використовуються як вхідні дані, на основі яких можливо розробити схему виконуваних операцій з цими даними.

Така архітектура TensorFlow дозволяє інтегрувати різноманітні моделі та алгоритми з областей машинного та глибокого навчання, включаючи нейронні мережі, забезпечуючи їх доступність через єдиний інтерфейс.

Це сприяє розробці діаграм потоків даних з вузлами обчислень, які виконують математичні дії. Кожен зв'язок між вузлами представляє багатовимірні вектори або матриці, формуючи тензори.

Попри те, що для взаємодії з TensorFlow зазвичай використовується Python, реальні математичні обчислення проводяться не на Python. Вони виконуються за допомогою ефективних виконавчих файлів, написаних на C++. Python в цьому контексті виконує роль посередника, що координує взаємодію між компонентами та забезпечує їх інтеграцію через високорівневі програмні абстракції.

Програми, створені на TensorFlow, можуть бути запуснені в різноманітних середовищах, включаючи мобільні пристрої на платформах iOS і Android, настільні

комп'ютери або навіть розподілені обчислювальні кластери в хмарі, використовуючи як центральні процесори, так і графічні процесори (або спеціалізовані процесори TPU від Google при роботі на платформі Google Cloud).

TensorFlow пропонує широкий спектр API, які охоплюють як високорівневі, так і низькорівневі інтерфейси. Google радить використовувати високорівневі API для спрощення процесу розробки даних та програм, тоді як низькорівневі API, відомі як TensorFlow Core, виявляються корисними для детального налагодження програм і проведення експериментів.

Модель програмування та основні поняття:

Кожне обчислення в TensorFlow описує орієнтований граф, який складається з вузлів і ребер, де вузли — це операції/функції, а ребра — переповнення вхідних і вихідних даних і ці функції.

- Входи/виходи в TensorFlow називаються Tensor. Тензор — це не що інше, як багатовимірний масив, для якого тип базового елемента вказується під час побудови графіка.
- Клієнтська програма, яка використовує TensorFlow, створює графіки потоку даних, використовуючи схвалені мови програмування (C або Python).
- Операція — це функція, яка має назву та представляє абстрактне обчислення. Операція може мати атрибути, які повинні бути згенеровані та передані під час побудови графіка.
- Одним із частих застосувань атрибутів є створення поліморфної операції.
- Ядро — це реалізація операції на конкретному пристрої.
- Клієнтська програма взаємодіє з входом системи TensorFlow шляхом створення сеансів. Інтерфейс сеансу має розширені стилі для створення графа обчислення та його метод підсилення `run()`, який обчислює вихідні дані окремого вузла в графі переслідування, надаючи необхідні вхідні дані.
- У більшості завдань машинного навчання обчислювальні графіки реалізуються багаторазово, і більшість звичайних тензорів не виживають після непідключеного виконання, тому TensorFlow має змінні.

- Змінна — це особливий вид операції, яка повертає дескриптор змінної Тензор, який може виживати під час багаторазових переслідувань графа.
- Параметри, які можна навчити, як-от ваги, зміщення, переміщуються в тензори, що зберігаються у змінних.

### **Гнучкість та Масштабованість**

Гнучкість та масштабованість TensorFlow є фундаментальними характеристиками, які роблять цю бібліотеку незамінною для розробки інтелектуальних систем, зокрема в сфері оцінювання якості мультимедійного контенту. Гнучкість TensorFlow проявляється у можливості легкої адаптації моделей під специфічні задачі шляхом зміни архітектури нейронних мереж, що дозволяє дослідникам та розробникам експериментувати з різноманітними конфігураціями для досягнення оптимальних результатів. Вони можуть керувати кількістю та розміром прихованих шарів, налаштовувати гіперпараметри, такі як швидкість навчання, та обирати різні функції активації для кожного нейрона.

Масштабованість TensorFlow демонструється у його здатності ефективно обробляти великі набори даних та складні архітектури моделей. TensorFlow оптимізовано для роботи з обчислювальними кластерами, що включають як CPU, так і GPU, а також з підтримкою розподілених обчислень на кількох пристроях. Це означає, що розробники можуть масштабувати свої проекти від прототипів на одному комп'ютері до великих систем, що обробляють мільйони або навіть мільярди точок даних на кластерах з високою продуктивністю.

### **Продуктивність та Швидкість**

Продуктивність та швидкість обчислень у TensorFlow безпосередньо впливають на ефективність тренування та валідації моделей глибинного навчання. Завдяки оптимізації для використання на GPU, TensorFlow забезпечує значне прискорення обчислень, що критично необхідно при роботі з великими обсягами мультимедійних даних та складними нейронними мережами [35]. GPU-оптимізовані обчислення дозволяють паралельну обробку даних, значно скорочуючи час, необхідний для тренування моделей, і роблять можливим використання більш глибоких та складніших архітектур.

Крім того, TensorFlow пропонує розширені можливості для оптимізації продуктивності, такі як автоматичне управління пам'яттю та динамічне розподілення ресурсів, що дозволяє максимально ефективно використовувати доступні обчислювальні ресурси. Це забезпечує розробникам гнучкість у виборі між швидкістю тренування та вимогами до точності моделі, дозволяючи досягти оптимального балансу залежно від потреб проекту.

### **Екосистема та Спільнота**

Екосистема TensorFlow є однією з найбільш розгалужених і активних у сфері машинного навчання та глибокого навчання. Ця екосистема включає в себе широкий спектр інструментів, бібліотек, та платформ, розроблених як офіційною командою TensorFlow, так і спільнотою незалежних розробників. Важливість екосистеми TensorFlow полягає не лише в різноманітності доступних інструментів, але й у підтримці спільноти, яка включає вчених, дослідників, розробників, і користувачів з усього світу. Спільнота обмінюється знаннями, досвідом, вирішує спільні проблеми, і сприяє швидкому впровадженню інноваційних рішень.

Активне спілкування у форумах, на GitHub, у блогах та на конференціях дозволяє розробникам знайти відповіді на свої питання, поділитися власними досягненнями, а також знайти партнерів для спільних проектів. Ця відкритість і колаборативність є ключовими чинниками швидкого розвитку технологій машинного навчання та глибокого навчання.

### **Візуалізація з TensorBoard**

TensorBoard є невід'ємною частиною екосистеми TensorFlow, що забезпечує потужні можливості візуалізації для аналізу та моніторингу процесу навчання моделей глибокого навчання. TensorBoard дозволяє користувачам візуально трасувати метрики навчання, такі як втрати (loss) та точність (accuracy), в реальному часі, переглядати архітектуру моделі, а також аналізувати розподіли та градієнти параметрів [37]. Така візуалізація надає безцінні інсайти про ефективність моделі, допомагає ідентифікувати проблеми у навчанні, такі як перенавчання або недонавчання, та сприяє більш швидкому та ефективному налагодженню.

Однією з унікальних можливостей TensorBoard є візуалізація високовимірних даних за допомогою техніки зниження розмірності, як-от t-SNE або PCA. Це дозволяє дослідникам бачити, як модель групує або розділяє дані в просторі ознак, що може бути корисним для зрозуміння, як модель сприймає різні класи або категорії даних.

### 2.2.3 Фреймворк Django

Django - це високорівневий фреймворк для розробки веб-додатків на мові програмування Python, який слідує архітектурному шаблону "модель-вид-контролер" (MVC). Він був розроблений для спрощення створення складних, даних-орієнтованих веб-сайтів, автоматизуючи чимало звичних задач, таких як адміністрування сайту (див. рис. 2.4), робота з базами даних, аутентифікація користувачів, і так далі [38].

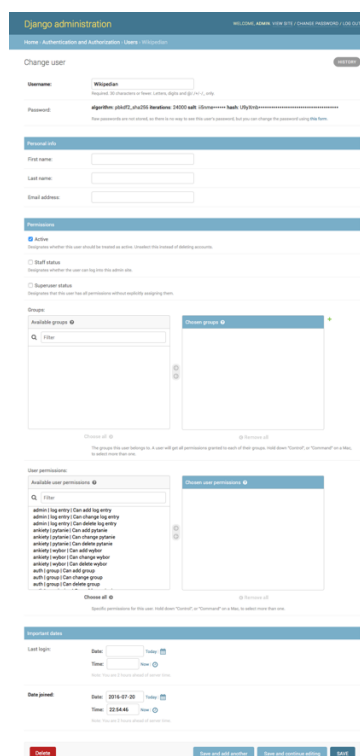


Рисунок 2.4 – Адміністративна панель Django

Використання Django обґрунтоване рядом причин:

1. Швидка розробка: Django дозволяє розробникам швидко переходити від концепції до запуску, завдяки великій кількості готових рішень, що йдуть "з коробки".

2. Багатофункціональність: Включає в себе систему адміністрування, форми, системи аутентифікації користувачів, мапінг об'єктно-реляційної бази даних (ORM) і багато іншого [40].

3. Безпека: Django забезпечує вбудовану захист від багатьох загроз безпеці веб, таких як SQL ін'єкції, cross-site scripting, cross-site request forgery та інше, допомагаючи розробникам уникнути багатьох поширених помилок безпеки.

4. Масштабованість: Фреймворк добре підходить для розробки як маленьких, так і великих проектів, здатний ефективно обробляти велику кількість трафіку і даних.

5. Велика спільнота та документація: Django має велику і активну спільноту розробників, а також обширну і детальну документацію, що робить легшим вивчення фреймворка та пошук допомоги при вирішенні проблем.

6. Гнучкість: Фреймворк дозволяє розробникам використовувати його для створення різноманітних типів веб-додатків, від простих веб-сайтів до складних веб-програм.

Завдяки цим перевагам, Django є популярним вибором серед розробників для створення веб-додатків на Python, допомагаючи їм більш ефективно працювати над проектами та створювати високоякісні, безпечні веб-сайти.

## **Висновки до розділу 2**

У процесі створення системи оцінювання мультимедійного контенту було використано ряд ключових платформ та інструментів, кожен з яких зіграв незамінну роль у розробці та імплементації проекту. Редактор коду Visual Studio Code слугував основним середовищем для написання та відлагодження коду, забезпечуючи зручність та ефективність розробки завдяки своїй гнучкості та багатофункціональності.

Python, як основа розробки, дозволив нам використовувати потужні бібліотеки та фреймворки, що сприяли створенню ефективної аналітичної системи. Бібліотека NumPy надала інструменти для виконання складних математичних та наукових обчислень, що є невід'ємною частиною обробки даних. TensorFlow, зі



своєю підтримкою алгоритмів машинного навчання та глибокого навчання, дозволило розробити та тренувати нейромережі для точного аналізу та оцінювання мультимедійного контенту. Нарешті, фреймворк Django був використаний для створення веб-додатку, що є інтерфейсом нашої системи, забезпечуючи масштабованість, безпеку та зручність у використанні.

Загалом, інтеграція цих платформ та інструментів дозволила створити комплексну систему оцінювання мультимедійного контенту, яка відрізняється високою точністю оцінок, гнучкістю у розробці та зручністю для кінцевого користувача. Цей досвід підкреслив значення вибору правильних технологій для досягнення цілей проекту та забезпечення високої якості розробленого продукту.

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ОЦІНКИ КОНТЕНТУ

#### 3.1 Архітектура аналітичної системи оцінювання

Для реалізації інтелектуальної системи оцінки контенту запропонована наступна архітектура. Вона представлена у вигляді діаграми класів. Вона складається з підсистеми яка зберігає дані, підсистеми аналізу та інтерфейсу.

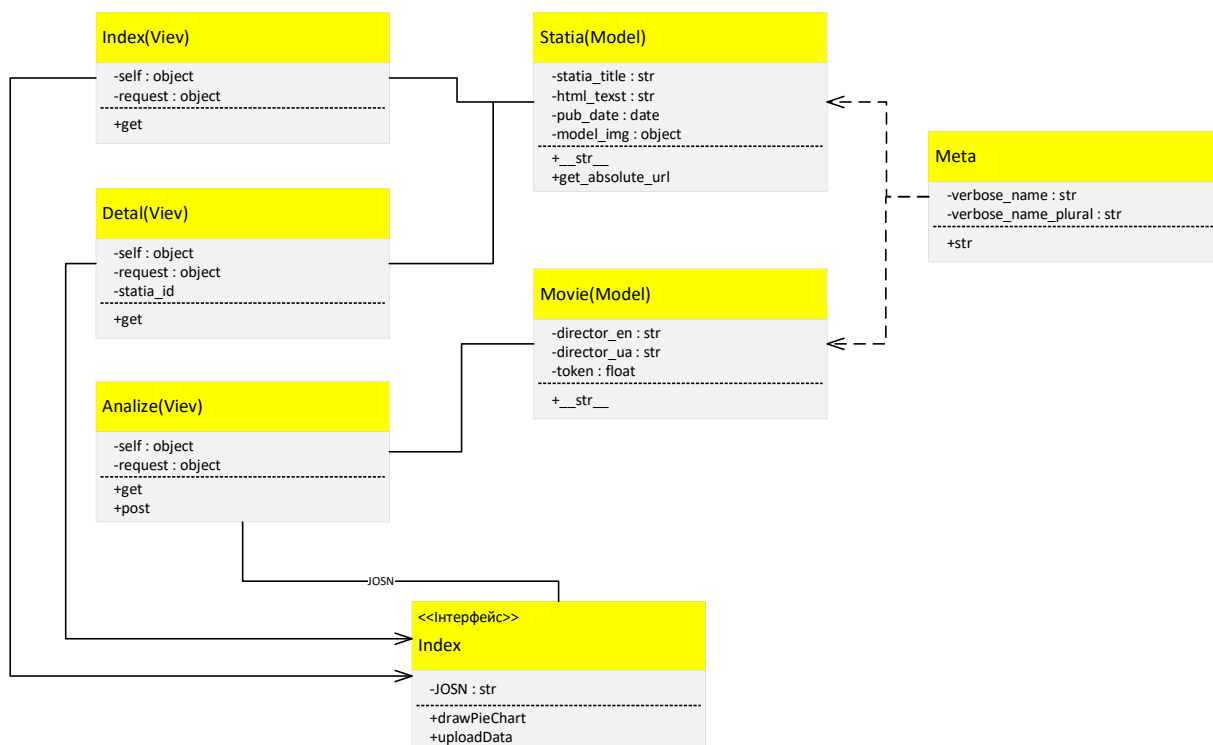


Рисунок 3.1 – (Class diagram) Діаграма класів додатка

На рисунку 3.1 представлено діаграму класів додатка. Класи Statia та Movie унаслідуються від загального класу Django – Model і відповідають за створення та зв'язок з базою даних. Атрибутами класу Movie є поля бази даних: ім'я режисерів та сценаристів англійською мовою (director\_en), ім'я режисера та сценариста українською мовою (director\_ua) та відповідний токен (token). Останнє поле має бути унікальним. Атрибутами класу Statia є назва статі (statia\_title), текст статі з html розміткою (html\_textst), дата публікації (pub\_date) та зображення (model\_img). Магічний метод \_\_str\_\_ відповідає за присвоєння зрозумілого ім'я класу, що

спрощує розробку. Метод `get_absolute_url` відповідає за формування абсолютної URL адреси до кожної зі статей блоку новин.

Клас `Meta` є вкладеним класом для кожної з моделей та відповідає за відображення назв блоків в панелі адміністрування сайту.

Усі класи відповідальні за обробку GET та POST запитів (`Index`, `Detail`, `Analyze`) унаслідуються від загального класу Django – `View`.

Клас `Index` відповідає за формування головної сторінки сайту, має єдиний метод `get` який відповідає за обробку запитів типу GET. Для формування заголовків новин клас має асоціативний зв'язок з моделлю даних відповідальну за таблицю новин (`Statia`).

Клас `Detail` відповідає за формування сторінок новин, тому має третій атрибут ідентифікатор статі (`statia_id`), який отримує з колекції `Get` GET запиту. З тих же причин, що і клас `Index`, клас `Detail` має асоціативний зв'язок з класом моделі `Statia`.

Клас `Analyze` та його метод `post` відповідають за обробку POST AJAX запиту від JS коду інтерфейсу, метод `get` є заглушкою, та викидає на Front-End повідомлення, що для цієї адреси методу GET не існує. Логіка методу `post` виконує розпаковування та перевірку даних отриманих у форматі JSON від користувацької частини, відшукує в базі даних токени відповідні веденим користувачем строковим даним. Для цього клас `Analyze` має асоціативний зв'язок з класом моделі `Movie`. Саме в методі `post` класу `Analyze` відбувається під'єднання нейромережових моделей, передача даних до них і отримання відповіді. Метод повертає на Front-End дані в форматі JSON, або код помилки.

За логіку роботи користувацької частини відповідають дві JavaScript функції: `drawPieChart` та `uploadData`. Друга відповідає за обробку форми: упаковує дані в формат JSON, виконує AJAX запит, отримує відповідь від сервера, розпаковує дані у форматі JSON, виводить передбачений рейтинг користувачу та викликає функцію `drawPieChart`, яка відповідає за відображення кругових діаграм, або видає користувачу інформацію про помилку.

Також пропоную більш детально роздивитись алгоритм роботи методу `post` класу `Analyze` (рис 3.2).

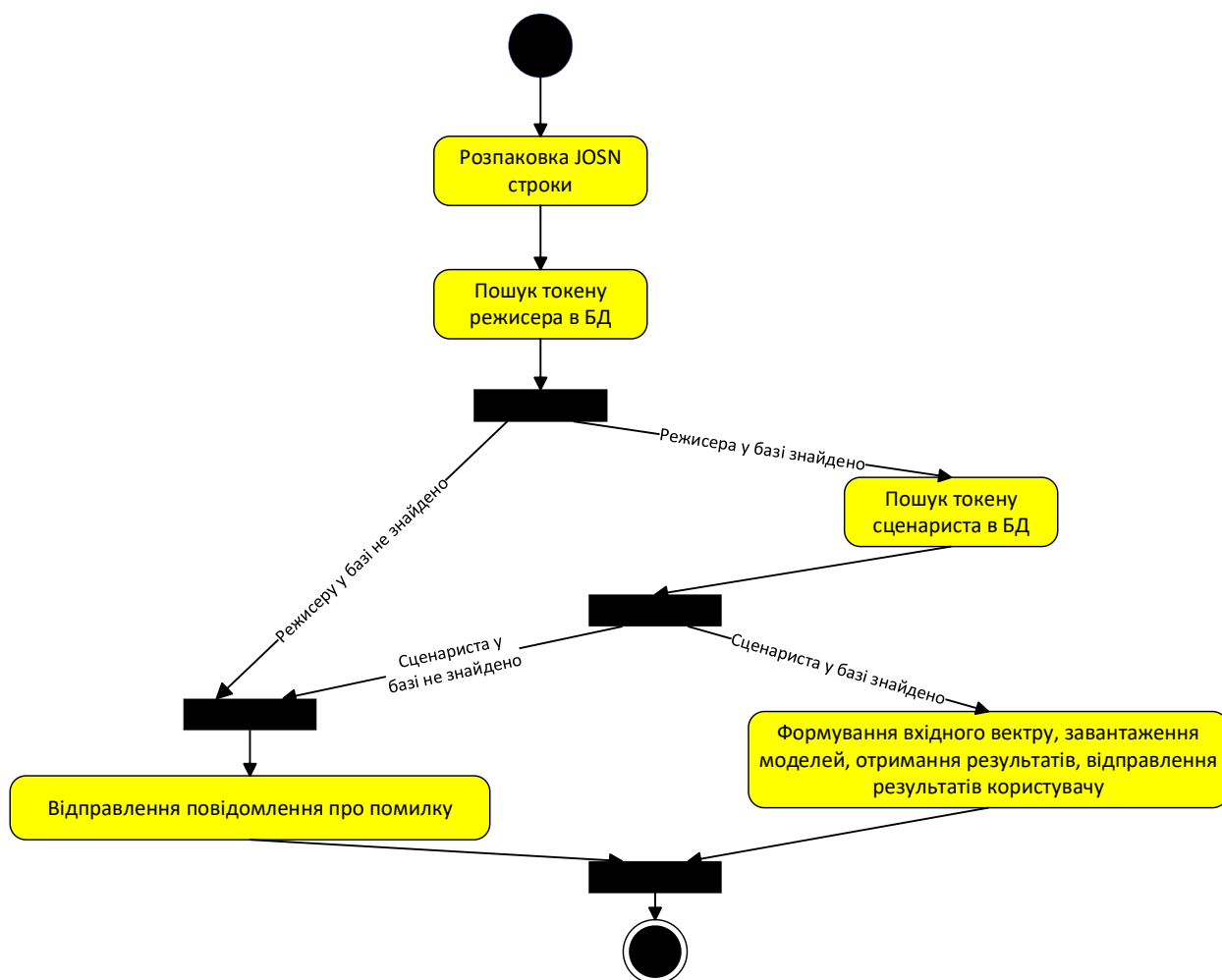


Рисунок 3.2 – (Activity diagram) Діаграма активності методу post класу Analyze

Для розпаковування даних з формату JOSN використовується стандартна бібліотека Python – json її метод loads та функція data. Для пошуку токена використовується стандартний менеджер моделі objects та його метод filter. Для створення вхідного вектора виконується нормалізація вікових обмежень, поділ строки з даними з обраних жанрів фільму у список строк за пробілом за допомогою методу split, а також перетворення списку строк у список чисел з плаваючою комою. Отриманий список перетворюється у двомірний вектор, зрозумілий нейромережевим моделям за допомогою методу array бібліотеки NumPy.

Після обробки даних нейронними мережами вектори відповідей NumPy перетворюються у зрозумілі JOSN формату списки Python, які обробляються з використанням функції map та лямда-функцій. Після чого оброблені списки

упаковуються в словник, який за допомогою функції Django JsonResponse перетворюються у JOSN формат.

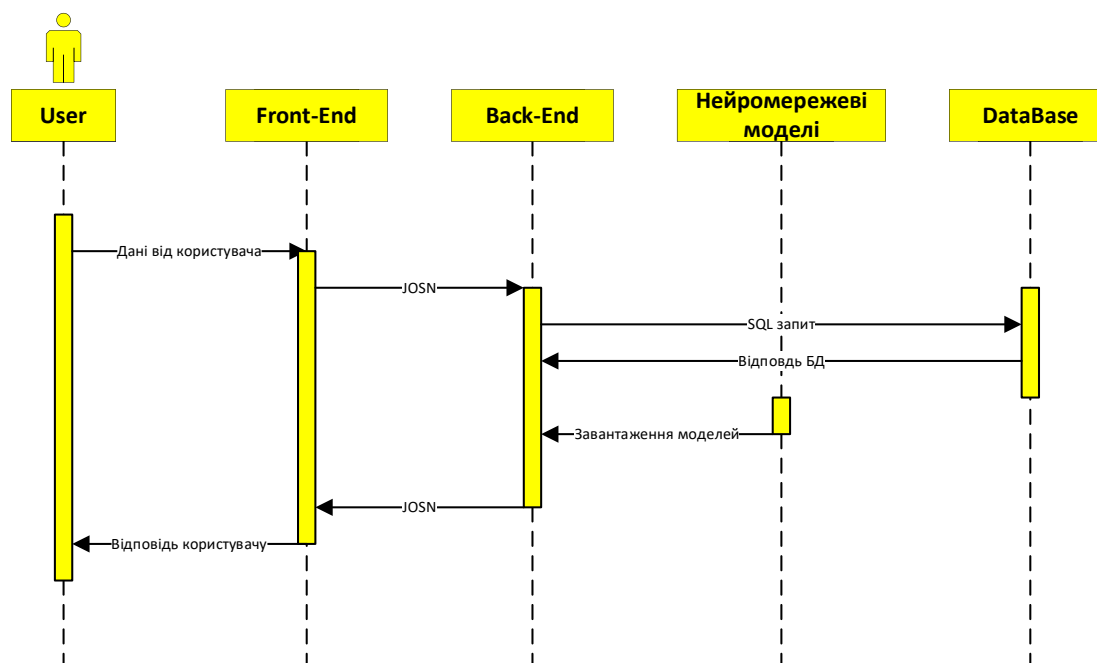


Рисунок 3.3 – (Sequence diagram) Діаграма послідовності роботи алгоритму передбачення аудиторії фільму

На діаграмі послідовності (див.рис.3.3) можна простежити потоки даних між користувацькою та серверною частинами додатка. Отримані дані від користувача обробляються на Front-End й упаковуються в формат JOSN, процес розпакування JOSN на сервері добре описано у діаграмі активності (рис 2). За формування SQL запиту відповідає ORM Django, за завантаження моделей метод load\_model класу models пакету keras. Процес формування зворотної відповіді користувачу також добре описано у діаграмі класів та діаграмі активності.

### 3.2 Використання технологій нейронних мереж в системах оцінювання контенту

Для навчання нейронних мереж було створено датасет шляхом комбінування даних з двох відкритих наборів даних які було знайдено у соціальній мережі для аналітиків та спеціалістів з машинного навчання – Kaggle [40].

Задачу створення нейронної мережі для розв’язання задачі: передбачення розподілу за статтю., розподілу за віковими групами, розподілу за професіями та передбачення рейтингу IMDb було розв’язано створенням окремих нейронних мереж для кожної з задач. У якості архітектури нейронної мережі було обрано перцептрон, бо саме ця архітектура гарно показує себе у задачах класифікації даних.

Вибір багат шарового перцептрону (БШП) (див.рис.3.4) як основного інструменту для розробки інтелектуальної системи оцінювання якості мультимедійного контенту впливає з низки його переваг та унікальних характеристик, які роблять його ідеально підходящим для вирішення задачі прогнозування та класифікації. Багат шаровий перцептрон, будучи фундаментальною моделлю в глибокому навчанні, демонструє вражаючу здатність до виявлення складних нелінійних зв'язків у великих наборах даних, що є критично важливим для аналізу та оцінки мультимедійного контенту [41].

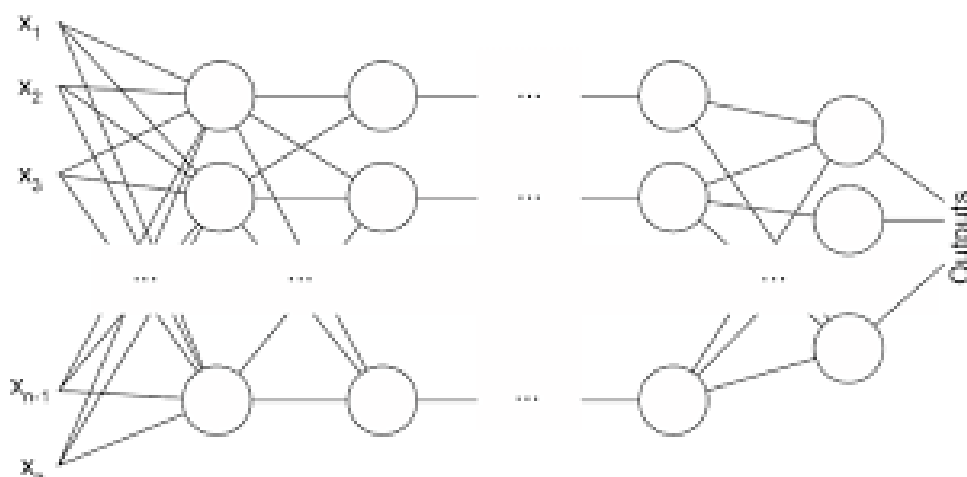


Рисунок 3.4 – Багат шаровий перцептрон

### Технічні аспекти та основні характеристики

Багат шаровий перцептрон складається з входного шару, одного або декількох прихованих шарів та виходового шару. Кожен нейрон у цих шарах з'єднаний з нейронами наступного шару з певною вагою, що оновлюється в процесі

навчання. Це дозволяє БШП адаптуватися до складних даних та виявляти взаємозв'язки між різними характеристиками контенту, використовуючи процес, заснований на градієнтному спуску та зворотньому поширенні помилки для оптимізації вагових коефіцієнтів.

### **Глибина аналізу та здатність до узагальнення**

Використання прихованих шарів надає БШП здатності виявляти та моделювати складні патерни в даних, що не завжди очевидні на поверхневому рівні. Це особливо корисно при оцінці мультимедійного контенту, де якість може залежати від низки суб'єктивних та об'єктивних факторів, таких як сюжетна лінія, акторська гра, звукове оформлення, візуальні ефекти та інші аспекти, які можуть взаємодіяти нелінійно. БШП ефективно враховує ці взаємодії, надаючи комплексне бачення якості контенту.

### **Адаптивність та гнучкість моделі**

Адаптивність та гнучкість багат шарового перцептронну (БШП) відіграють ключову роль у розробці ефективних систем оцінювання якості мультимедійного контенту. Ці характеристики дозволяють моделі точно налаштуватися під конкретні потреби та особливості задач, з якими вона стикається. Адаптивність БШП до різноманітних даних і завдань досягається через можливість зміни кількості прихованих шарів і нейронів в кожному шарі, що дозволяє моделювати як прості, так і вкрай складні взаємозв'язки між входовими даними і цільовими змінними.

Гнучкість БШП також проявляється в здатності ефективно працювати з різними типами даних, що є особливо важливим при аналізі мультимедійного контенту, де одночасно можуть використовуватися текст, зображення, аудіо та відео. Використання технік передобробки та векторизації даних, таких як one-hot encoding для категоріальних даних або використання вбудовувань (embeddings) для текстової інформації, дозволяє перетворювати різнорідні дані в формат, з яким може працювати БШП, забезпечуючи глибокий аналіз та високу точність прогнозування.

### **Переваги в контексті мультимедійного контенту**

У контексті мультимедійного контенту багатосаровий перцептрон відкриває широкі можливості для аналізу та оцінки якості. Однією з ключових переваг є здатність БШП виявляти та аналізувати нелінійні зв'язки між різноманітними характеристиками контенту, включаючи візуальні та аудіальні елементи, текстовий контекст, жанрові особливості, та інші атрибути, які впливають на сприйняття якості кінцевими користувачами.

Використання БШП дозволяє також автоматизувати процеси ідентифікації стилістичних, жанрових та тематичних елементів у мультимедійному контенті, що сприяє більш точній класифікації та оцінці матеріалів. Такий підхід може виявляти не тільки загальні тенденції, але й тонкі нюанси, які можуть впливати на оцінку якості, забезпечуючи більшу об'єктивність та глибину аналізу.

Окрім того, БШП здатний працювати з великими обсягами даних, що є важливим для сучасної індустрії мультимедійного контенту, де щодня генерується велика кількість нового матеріалу. Ефективна обробка та аналіз цих даних з використанням БШП може значно підвищити якість оцінювання та рекомендацій, забезпечуючи краще розуміння потреб та переваг аудиторії.

### **Інтеграція з іншими технологіями машинного навчання**

Окрім власних переваг, багатосаровий перцептрон може ефективно інтегруватися з іншими технологіями та методами машинного навчання, зокрема, з алгоритмами згорткових нейронних мереж (ЗНМ) для обробки зображень та відео, а також з рекурентними нейронними мережами (РНМ) для аналізу послідовностей даних, як-от аудіо та текст. Ця синергія дозволяє створювати комплексні системи оцінювання якості, які використовують переваги кожного підходу для розгляду різних аспектів мультимедійного контенту, забезпечуючи більш глибоке та всебічне розуміння якості.

### **Виклики та майбутні перспективи**

Попри численні переваги, використання багатосарового перцептронного також ставить перед дослідниками та розробниками певні виклики. Одним з основних є необхідність великої кількості даних для ефективного тренування моделі, що може бути обмежувальним фактором для деяких додатків. Крім того, процес



налаштування моделі, включаючи вибір архітектури та гіперпараметрів, вимагає глибокого розуміння як самої моделі, так і специфіки задачі, над якою працює система.

Незважаючи на ці виклики, постійний розвиток технологій та методів машинного навчання відкриває нові перспективи для подальшого удосконалення багатосарових перцептронів та розширення їх застосування. Дослідження в області оптимізації тренування, зниження вимог до обсягів даних та автоматизації вибору архітектури можуть подолати існуючі бар'єри та зробити БШП ще більш доступним та ефективним інструментом для аналізу мультимедійного контенту.

У сукупності, багатосаровий перцептрон є потужним інструментом для розробки інтелектуальних систем оцінювання якості мультимедійного контенту, пропонуючи глибокий аналітичний потенціал, гнучкість та адаптивність. Його здатність інтегруватися з іншими методами машинного навчання, виклики, що стоять перед дослідниками, та майбутні перспективи розвитку лише підкреслюють значення та актуальність БШП у сучасній науковій та технологічній арені, відкриваючи нові шляхи для інновацій у галузі мультимедіа.

Кожна мережа має 22 входи на які подаються наступні дані: ім'я режисера, ім'я сценариста, вікові обмеження, з 4 по 22 входи (19 шт відповідають за жанри фільму та мають тільки два стани 0 та 1, кожен фільм має щонайменше один жанр (невідомий, вистава, пригоди, анімація, дитячий, комедія, детектив, документальний, драма, фентезі, фільм-нуар, жахи, мюзикл, таємниці, романтика, наукова фантастика, трилер, війна, вестерн)).

Для збереження та обробки даних використовувався табличний редактор MS Excel та власноруч написані скрипти на Python з використанням бібліотеки Openruhl.

Для створення вхідного вектора було виконано токенізацію для текстових даних ім'я режисера та ім'я сценариста за допомогою вбудованого в пакет keras інструменту – класу Tokenizer. Для даних вікових обмежень та токенів було виконано нормалізацію даних в діапазоні від 0 до 1. Вхідний список було перетворено у вектор за допомогою бібліотеки NumPy.

Для створення нейронних мереж було обрано фреймворк TensorFlow, який було розроблено командою Google Brain з метою забезпечення високої продуктивності та гнучкості у розробці та використанні глибоких нейронних мереж. Та Keras у якості інтерфейсу високого рівня дозволяє зменшити кількість коду для створення та навчання моделей, спрощуючи процес розробки та забезпечуючи більшу зрозумілість та читабельність коду. Такий підхід дозволив хід розробки та дозволив отримати більш якісні моделі.

Нейромережа для передбачення розподілу аудиторії за віком має два скритих шари, функція активації – ReLu, для вихідного шару – Softmax. Мережа має 540 параметрів, для роботи яких знадобиться 2,11 КВ оперативної пам'яті. Докладніше див. таблицю 3.1. У якості функції втрат використовувалася категоріальна кросентропія, оптимізатор adam. Експериментальним шляхом було визначено, що оптимально навчання проводити у 10 епох, що забезпечує точність для варіаційної вибірки у 87%.

Вихідний вектор складається з чотирьох значень, які вказують розподіл майбутньої аудиторії у відсотках за чотирма групами: до 24 років, від 25 до 35 років, від 36 до 49 років, старше 50 років.

Таблиця 3.1 – Структура нейромережі

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 16)	368
dense_1 (Dense)	(None, 8)	136
dense_2 (Dense)	(None, 4)	36

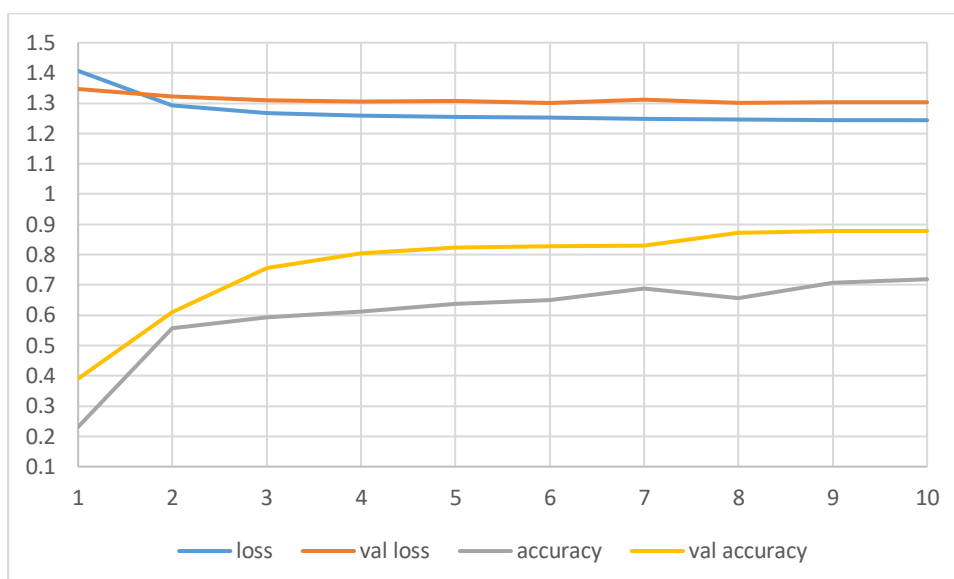


Рисунок 3.5 – Графік навчання нейронної мережі для передбачення розподілу за віком

Нейромережа для передбачення розподілу аудиторії за статтю має один скритий шар, функція активації – ReLu, для вихідного шару – Softmax. Мережа має 402 параметри, для роботи яких знадобиться 1,57 КВ оперативної пам'яті. Докладніше див. таблицю 3.2. У якості функції втрат використовувалася категоріальна кросентропія, оптимізатор adam. Експериментальним шляхом було визначено, що оптимально навчання проводити у 10 епох, що забезпечує точність для варіаційної вибірки у 97% (див.рис. 3.6).

Вихідний вектор складається з двох значень.

Таблиця 3.2 – Структура нейромережі

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 16)	368
dense_1 (Dense)	(None, 2)	34

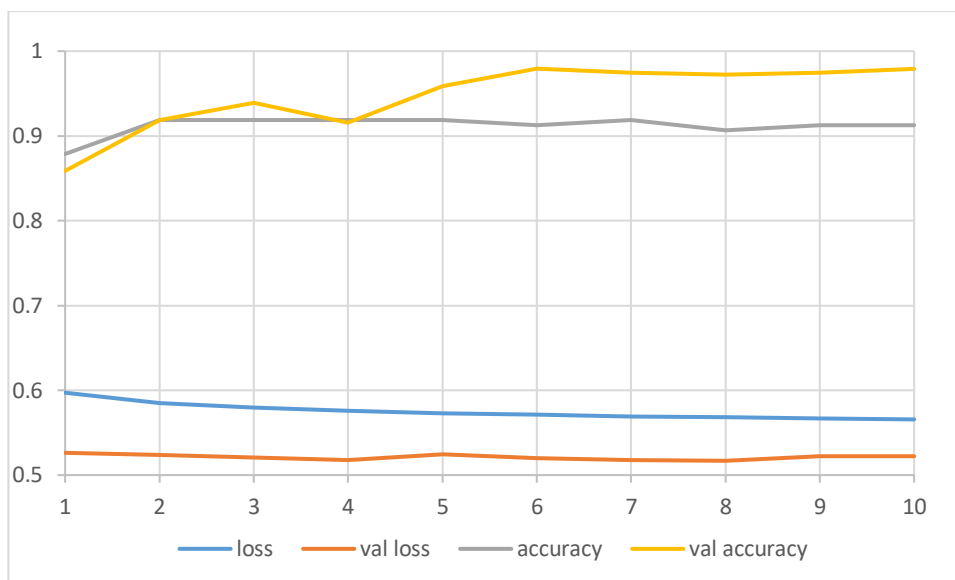


Рисунок 3.6 – Графік навчання нейронної мережі для передбачення розподілу за статтю

Нейромережа для передбачення розподілу аудиторії за професією має один скритий шар, функція активації – ReLu, для вихідного шару – Softmax. Мережа має 1781 параметр, для роботи яких знадобиться 6,96 КВ оперативної пам'яті. Докладніше див. таблицю 3.3. У якості функції втрат використовувалася категоріальна кросентропія, оптимізатор adam. Експериментальним шляхом було визначено, що оптимально навчання проводити у 10 епох, що забезпечує точність для варіаційної вибірки у 78% (див.рис. 3.7).

Вихідний вектор складається з 21 значення, які вказують розподіл майбутньої аудиторії у відсотках за 21 групою: адміністратор, митець, лікар, освітянин, інженер, розважальник, керівник, медичний працівник, домогосподар, адвокат, бібліотекар, маркетолог, відсутній, інший, програміст, пенсіонер, продавець, вчений, студент, технік, письменник.

Таблиця 3.3 – Структура нейромережі

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 40)	920
dense_1 (Dense)	(None, 21)	861

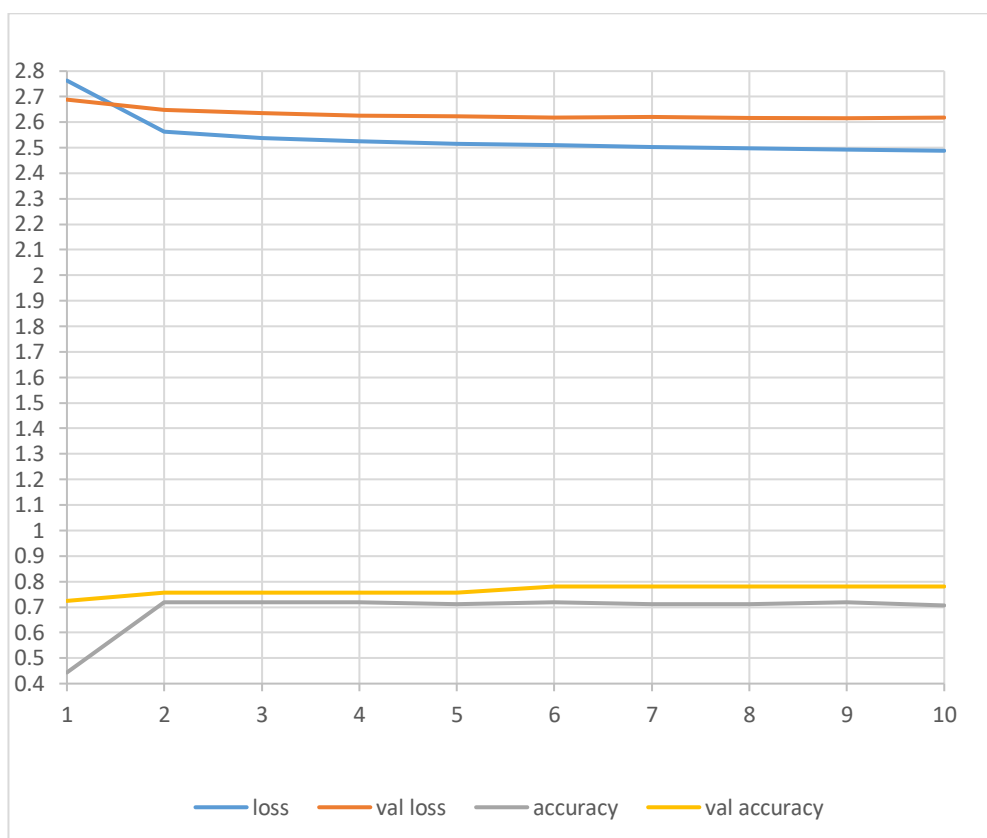


Рисунок 3.7 – Графік навчання нейронної мережі для передбачення розподілу за професіями

Нейромережа для передбачення рейтингу IMDb має один скритий шар, функція активації – ReLu, для вихідного шару – сигмоїда. Мережа має 241 параметр, для роботи яких знадобиться 0,96 КВ оперативної пам'яті. Докладніше див. таблицю 3.4. У якості функції втрат використовувалася середня абсолютна похибка, оптимізатор adam. Експериментальним шляхом було визначено, що оптимально навчання проводити у 15 епох, що забезпечує точність для варіаційної вибірки у 89% (див.рис. 3.8).

Вихідний вектор складається з одного значення – рейтингу IMDb.

Таблиця 3.4 – Структура нейромережі

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 10)	230
dense_1 (Dense)	(None, 1)	11

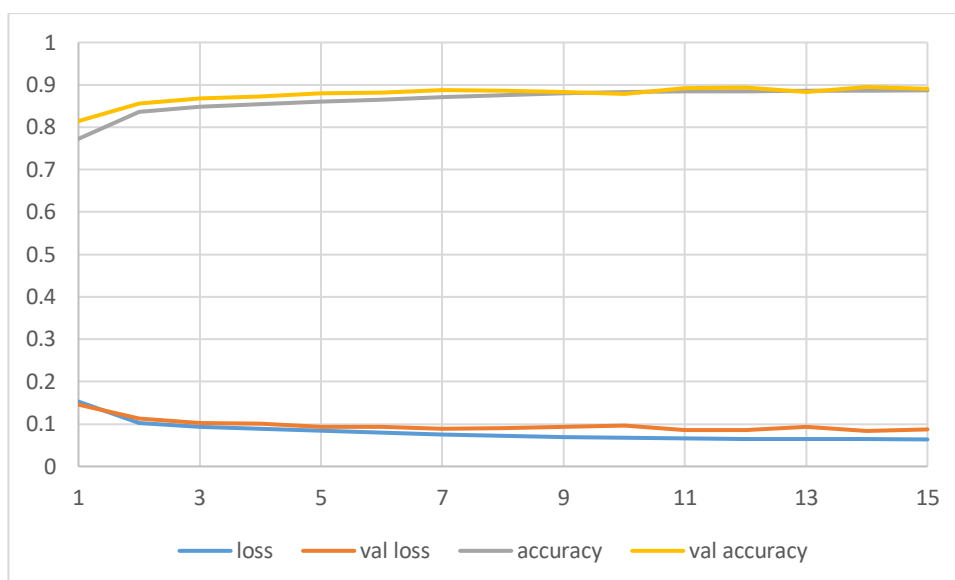


Рисунок 3.8 – Графік навчання нейронної мережі для передбачення рейтингу IMDb

Таким чином було розроблено 4 моделі які будуть використані у web додатку системи підтримки прийняття рішень для визначення цільової аудиторії фільму – «кіноаналізатор», який дозволить користувачам мати швидкий доступ до навчених моделей.

### 3.3 Розробка веб-додатку

Для розробки Web додатку було обрано фреймворк Django. Django - це веб-фреймворк, написаний на мові програмування Python з відкритим вхідним кодом, який використовується для швидкої та ефективної розробки веб-додатків. Django надає розробникам зручний спосіб визначення URL-адрес, оброблення HTTP-запитів, керування базами даних, створення шаблонів для відображення даних і безліч інших корисних функцій. Він також має вбудовану систему аутентифікації користувачів, адміністративну панель і набір інструментів для забезпечення безпеки додатків. Django є одним з найпопулярніших фреймворків для розробки веб-додатків у світі Python.

До того ж послідовне та акуратне кодування відповідно до документації Django робить проєкт досить легко масштабованим, що дозволить, за потреби,

розширювати проєкт майбутньому або використовувати його частини в інших проєктах.

Для передачі даних між користувацькою та серверною частиною використовується технологія AJAX запитів, яка дозволяє непомітно для користувача передати дані на сервер та отримати відповідь. Для побудови кругових діаграм використовувалася бібліотека chart.js. Бібліотека jQuery забезпечує зручність розробки та робить код більш компактним.

Для інформаційного забезпечення користувачів сайт оснащено невеличким блогом, де розмінюються новини проєкту.

Верстка сайту відповідає сучасним стандартам та забезпечує доступ до ресурсу з будь-яких пристроїв.

Сайт зустрічає користувача інформаційним повідомленням, після якого відразу знаходиться інтуїтивно зрозуміла форма для введення даних та кнопка «аналізувати». Всі операції з аналізу та обробки даних відбуваються з боку сервера. Після натискання кнопки дизайн сайту змінюється (див. рис. 3.9, 3.10) і перед користувачем з'являються кругові діаграми. Під час аналізу користувач може виключати деякі дані з діаграми шляхом кліка мишкою на їх мітку у легенді (рис 3.11). При наведенні мишкою на елемент діаграми користувач має можливість побачити назву категорії та відсоток.

Через те що неймережі вміють працювати тільки з відомими їм режисерами та сценаристами з боку сервера відбувається заміна текстових даних на відповідний їм токен і якщо в базі даних знайти відповідне ім'я не вдалося сервіс повідомить про це користувачу (рис 3.12).

## Кіноаналізатор

Проект Кіноаналізатор - це інноваційний інструмент, розроблений з використанням передових технологій штучного інтелекту та нейронних мереж. Його основна мета - надати кіноманам, кінокритикам та індустрії розваг засоби для більш глибокого розуміння та передбачення успішності фільмів. Завдяки аналізу аудиторії та рейтингу IMDb, Кіноаналізатор дозволяє прогнозувати прийняття фільму глядачами та його комерційний успіх.

За допомогою нейронних мереж та аналізу великих обсягів даних про попередні фільми, Кіноаналізатор здатний виявляти закономірності у смаках глядачів, популярних жанрах, акторських складах та інших факторах, які впливають на рейтинг фільмів. Це дозволяє студіям та режисерам заздалегідь прогнозувати реакцію аудиторії на їхні майбутні творіння та адаптувати свою стратегію маркетингу.

Будучи потужним інструментом для аналізу даних, Кіноаналізатор також надає корисну інформацію для кінематографічних критиків та дослідників, допомагаючи їм розуміти та аналізувати тренди в світі кіно та телевізійних шоу. Власники прав на фільми можуть використовувати дані, отримані від Кіноаналізатора, для вдосконалення своїх продуктів та визначення найефективніших стратегій просування на ринку. Кіноаналізатор відкриває нові можливості для індустрії розваг, забезпечуючи більш точний та передбачуваний аналіз кінофільмів.

Режесер:

Сценарист:


Вікові обмеження:

Жанри


<input type="checkbox"/> невідомий	<input type="checkbox"/> Документальний	<input type="checkbox"/> Романтика
<input type="checkbox"/> Вистава	<input type="checkbox"/> Драма	<input type="checkbox"/> Наукова фантастика
<input type="checkbox"/> Пригоди	<input type="checkbox"/> Фентезі	<input type="checkbox"/> Трилер
<input type="checkbox"/> Анімація	<input type="checkbox"/> Фільм-Наур	<input type="checkbox"/> Війна
<input type="checkbox"/> Дитячий	<input type="checkbox"/> Жахи	<input type="checkbox"/> Вестрн
<input type="checkbox"/> Комедія	<input type="checkbox"/> Мюзикл	
<input type="checkbox"/> Детектив	<input type="checkbox"/> Тьємниці	

[АНАЛІЗУВАТИ](#)


## Новини проєкту



База даних оновлено



Виправлення помилок



Початок проєкту

© 2024 Для дипломної роботи магістра

Рисунок 3.9 – Зовнішній вигляд сайту



## Кіноаналізатор

Проект Кіноаналізатор - це невіддільний інструмент розроблений з використанням передових технологій штучного інтелекту та нейронних мереж. Його основна мета - надати кіноманам, кінорежисерам та студіям розумні заходи для більш глибокого розуміння та корисних уявлень щодо фільмів. Завдяки аналізу аудиторії та рейтингу IMDb, Кіноаналізатор дозволяє прогнозувати успішність фільму створеного та його комерційний успіх.

За допомогою нейронних мереж та аналізу великих обсягів даних про популярні фільми, Кіноаналізатор здатний виявляти закономірності в смаках глядачів, популярних жанах, авторських стилях та інших факторах, які впливають на рейтинг фільмів. Ці дані використовуються для прогнозування успішності фільмів, призначення рецензій на веб-сайтах, рекомендацій та адаптувати свої стратегії маркетингу.

Дякую за увагу та інтерес до нашого проекту. Кіноаналізатор - це лише перший крок у нашій подорожі до більш глибокого розуміння та використання даних, отриманих від кіноаналізатора. Для покращення наших прогнозів та рекомендацій, ми плануємо використовувати штучний інтелект та машинне навчання для аналізу великих обсягів даних, отриманих від кіноаналізатора. Дякую за увагу та інтерес до нашого проекту. Кіноаналізатор - це лише перший крок у нашій подорожі до більш глибокого розуміння та використання даних, отриманих від кіноаналізатора.

**Рейсери:**

**Самефективність:**

**Високе об'єктивність:**

**Жанри:**

<input type="checkbox"/> невідомий	<input type="checkbox"/> Документальний	<input type="checkbox"/> Романтика
<input type="checkbox"/> Битва	<input checked="" type="checkbox"/> Драма	<input type="checkbox"/> Наукова фантастика
<input checked="" type="checkbox"/> Пригоди	<input type="checkbox"/> Фентезі	<input type="checkbox"/> Трилер
<input type="checkbox"/> Анімація	<input type="checkbox"/> Фільм-мюзикл	<input type="checkbox"/> Війна
<input type="checkbox"/> Дитячий	<input type="checkbox"/> Жази	<input type="checkbox"/> Вестри
<input type="checkbox"/> Комедія	<input type="checkbox"/> Мюзикл	
<input type="checkbox"/> Детектив	<input checked="" type="checkbox"/> Таскиниці	

[АНАЛІЗУВАТИ](#)

### Передбачені данні


**Рейтинг IMDb: 8.23**

**Розподіл за полами**


**Розподіл за віком**

**Розподіл за професіями**


### Новини проєкту



База даних оновлено



Виправлення помилок



Початок проєкту

© 2024 для дипломної роботи магістра

Рисунок 3.10 – Зовнішній вигляд сайту (виведення результатів)



Рисунок 3.11 – Можливість виключення даних з діаграми

Сценарист:  
uncloud

Вікові обмеження:  
16+

Жанри

<input type="checkbox"/> невідомий	<input type="checkbox"/> Документальний	<input type="checkbox"/> Романтика
<input type="checkbox"/> Вистава	<input checked="" type="checkbox"/> Драма	<input type="checkbox"/> Наукова фантастика
<input checked="" type="checkbox"/> Пригоди	<input type="checkbox"/> Фентезі	<input type="checkbox"/> Трилер
<input type="checkbox"/> Анімація	<input type="checkbox"/> Фільм-Наур	<input type="checkbox"/> Війна
<input type="checkbox"/> Дитячий	<input type="checkbox"/> Жахи	<input type="checkbox"/> Вестрн
<input type="checkbox"/> Комедія	<input type="checkbox"/> Мюзикл	
<input type="checkbox"/> Дедиктив	<input checked="" type="checkbox"/> Таємниці	

**АНАЛІЗУВАТИ**

Режесера або сценариста у базі не знайдено

Рисунок 3.12 – Відображення помилок

У якості демонстрації пропоную порівняти відповідь нейронної мережі з реальними даними: наприклад фільм 1996 року «Таблетка радості» реальний рейтинг IMDb 6,9, передбачений 7,08 (похибка склала 2%), а ось похибка в розподілі за статтю склала приблизно 20%, середня похибка в розподілі за віком склала 26%. Для фільму 1995 року «Аполлон 13» похибка в передбаченні рейтингу склала – 11%, в розподілі за статтю – 5%, в розподілі за віковими категоріями – 10,5%. Для мультфільму «Аладін» 1992 року похибка передбачення рейтингу склала – 10%, розподілу по статті – 1%, розподілу по віку – 5%.

Таким чином можна стверджувати, що заявлені похибки нейромереж відповідають дійсності, а розроблений додаток повністю відповідає технічному завданню та виконує всі свої функції. Запропоновані архітектури нейромереж ефективно оцінюють мультимедійний та кінематографічний контент.

### **3.4 Мобільна адаптація веб-додатку**

Мобільна адаптація веб-застосунку програми для оцінювання якості мультимедійного контенту є критично важливою з кількох причин, що визначають ефективність взаємодії з кінцевими користувачами та досягнення поставлених цілей. Розглянемо ключові аспекти, які підкреслюють необхідність мобільної адаптації:

#### **1. Зростаюча популярність мобільних пристроїв**

У сучасному світі спостерігається стрімке зростання популярності мобільних пристроїв, які стають основним засобом доступу до інтернету для більшості користувачів по всьому світу. Смартфони та планшети використовуються не лише для спілкування та соціальних мереж, але й для перегляду новин, читання книг, слухання музики та перегляду відеоконтенту. Це змінило звички споживання контенту, роблячи мобільні пристрої важливим каналом для доступу до мультимедійного контенту.

Для проекту оцінювання якості мультимедійного контенту адаптація під мобільні пристрої означає не тільки збільшення потенційної аудиторії, але й можливість забезпечити користувачам зручний доступ до оцінок, рецензій та

аналітики незалежно від того, де вони знаходяться. Мобільна адаптація дозволяє користувачам швидко отримувати інформацію про якість контенту прямо зі свого смартфона, підвищуючи залученість та задоволеність від використання застосунку.

## **2. Покращення користувацького досвіду**

Адаптація веб-застосунку під мобільні пристрої має безпосередній вплив на користувацький досвід. Вона забезпечує оптимальне відображення контенту на екранах різних розмірів, забезпечуючи легкість читання тексту, доступність інтерактивних елементів та зручність навігації. Важливою є також швидкість завантаження сторінок, яка часто є вирішальним фактором у залученні та утриманні користувачів. Мобільно-адаптований застосунок повинен бути оптимізований для ефективної роботи з мобільним інтернетом, що має нижчу пропускну спроможність порівняно зі стаціонарними з'єднаннями.

Покращення користувацького досвіду також включає інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам легко знаходити потрібну інформацію та виконувати потрібні дії, такі як оцінка контенту, читання рецензій або поділ з іншими. Адаптація під мобільні пристрої передбачає розробку елементів управління, оптимізованих для сенсорних екранів, та форматування контенту таким чином, щоб максимально використовувати обмежений простір екрану.

## **3. Підвищення доступності контенту**

Мобільна адаптація значно підвищує доступність контенту, роблячи його доступним для ширшої аудиторії в будь-який час і з будь-якого місця. У сучасному світі, де користувачі прагнуть отримувати інформацію на ходу, забезпечення зручного доступу до мультимедійного контенту через мобільні пристрої є ключовим фактором задоволення їхніх потреб. Важливість цього аспекту зростає у контексті оцінювання мультимедійного контенту, де швидке та зручне отримання оцінок, рецензій та аналітичних даних може впливати на рішення користувачів щодо перегляду або покупки контенту.

## **4. Збільшення відвідуваності та залучення аудиторії**

Мобільна адаптація веб-застосунку відкриває двері для збільшення відвідуваності та залучення нової аудиторії. З огляду на зростання кількості

користувачів мобільного інтернету, адаптовані під мобільні пристрої сайти отримують вищу позицію в результатах пошукових систем, що сприяє збільшенню органічного трафіку. Крім того, мобільно-оптимізовані застосунки мають кращі шанси на поширення через соціальні мережі, оскільки користувачі часто діляться контентом, який легко переглядати на мобільних пристроях. Це забезпечує залучення більшої кількості користувачів та сприяє формуванню лояльної аудиторії.

### **5. Підтримка сучасних стандартів веб-розробки**

Адаптація під мобільні пристрої є не лише питанням зручності користувачів, але й відповідністю до сучасних стандартів веб-розробки. Сучасні веб-стандарти та кращі практики, такі як Responsive Web Design (RWD), передбачають створення веб-застосунків, які автоматично адаптуються під різні розміри екранів та орієнтації. Це не тільки покращує загальний досвід користувача, але й сприяє підвищенню рейтингу сайту в пошукових системах, оскільки мобільна адаптація є одним з факторів ранжування. Відповідність сучасним стандартам веб-розробки також підкреслює професіоналізм розробників і демонструє відданість наданню якісних та актуальних рішень для кінцевих користувачів.

У підсумку, мобільна адаптація веб-застосунку для оцінювання якості мультимедійного контенту не просто додає зручність для користувачів, але є стратегічно важливим кроком, що відкриває широкі можливості для збільшення відвідуваності, покращення користувацького досвіду, залучення нової аудиторії та підтримки високих стандартів веб-розробки.

Для адоптації сайту було використано метод так званої гумової верстки. Гумова верстка – це метод веб-розробки який передбачає використання відносних одиниць вимірювання, що дозволяє розмірам підлаштовуватись під параметри екрану. Зрозуміло, що розміри далеко не всіх елементів сторінки можуть бути вказані у відносних одиницях: значення кеглю шрифту, розміри графічних елементів, деякі відступи інколи досить складно визначити у відносних одиницях. До того ж адаптивність сайту передбачає не тільки зміну розмірів, а перебудову

контенту з горизонтального в вертикальне положення. Тому для забезпечення адаптивності використовуються медіазапити CSS.

У власному проєкті мною було використано лише один медіазапит для екранів розмір яких менше 1000px по горизонталі (під цю характеристику підпадають всі наразі існуючі смартфони). Так чекбокси в формі з трьох колонок перебудовуються в одну, блоки заголовків новин займають всю ширину сторінки підлаштовуючи форму малюнку шляхом обрізання. Збільшується значення кеглю шрифтів. Також зменшується відносно значення бокових відступів блоків з контентом з 15% на сторону до 1% на сторону.

Але елементи Chart.js (діаграми) налаштовуються з JavaScript тому, щоб зробити їх адаптивними знадобилося зробити перевірку розміру екрану на рівні JavaScript. До того ж було створено два окремих словника налаштувань для мобільної (option\_dict\_mobile) та комп'ютерної (option\_dic) версій.

```
var option_dict_mobile = {
  plugins: {
    tooltip: {
      titleFont: {size: 40},
      bodyFont: {size: 30}},
    legend: {
      labels: {
        font: {size: 40, color: '#000000'}
      }}}
var option_dict = {
  plugins: {
    tooltip: {
      titleFont: {size: 14},
      bodyFont: {size: 12}},
    legend: {
      labels: {
        font: {size: 14, color: '#000000'}
      }}}
}
```

Детальніше зміни елементів для адаптації до мобільної версії сайту описано в таблиці 3.5.

Таблиця 3.5 – Порівняння властивостей CSS у десктопній та мобільній версії

Клас або тег	Властивість	десктопна	мобільна
conten_container	width	70%	98%
container	justify-content	center	space-between
diogram	width	600px	100%
block-kon	display	flex	block
h5	font-size	20px	40px
h3	font-size	30px	50px
label	font-size		40px
Input [type="text"], select,	height		80px
	font-size		40px
input[type ="checkbox"]	width	20px	30px
	height	20px	30px
	display	inline	inline-block
	margin-right		20px
btn	margin		auto
	width	300px	80%
	height	50px	100px
	font-size	20px	40px
	padding	10px 20px	20px
	flex-direction		column
	align-items		center
	margin-top		40px
	margin-left	50%	0%
checkbox	width		98%
	float		left

Клас або тег	Властивість	десктопна	мобільна
new_text	justify-content		center
	padding-left		10px
	padding-right		5px
block_img	width	300px	98%
	height	300px	auto
block_new	width	300px	98%
new_img	width	400px	1000px

При запуску веб-застосунку на мобільному пристрої, користувачі бачать зручно структурований початковий екран, де вони можуть ввести або вибрати необхідні параметри для аналізу мультимедійного контенту (див. рис. 3.13):

- Вибір режисера: Текстове вікно, що дозволяє користувачу легко знайти та обрати інтересуючого режисера.

- Вибір сценариста: Аналогічне текстове вікно для сценаристів, що сприяє деталізації пошуку.

- Вибір жанру: Мультиселектор з можливістю обрати один або декілька жанрів для більш точного аналізу.

- Вибір вікового обмеження: Випадаючий список для встановлення вікового цензу, що дозволяє адаптувати пошук під вікові вподобання.

- Кнопка "Аналізувати": Велика та яскрава кнопка на початковому екрані, яка ініціює процес аналізу на основі обраних параметрів.

Після натискання на кнопку "Аналізувати", система обробляє запит та відображає результати на новому екрані, організованому для зручного перегляду на мобільних пристроях:

- Рейтинг IMDb: Відображення загального рейтингу IMDb для аналізованого контенту, представлене у вигляді числа або зіркової оцінки.



- Розподіл за статтю: Діаграма, що відображає вподобання контенту серед різних статей.

- Розподіл за віком: Діаграма, що показує розподіл вікових груп серед аудиторії контенту (див.рис. 3.14).

- Розподіл за професіями: Візуалізація, яка демонструє, як різні професійні групи оцінюють контент (див.рис.3.15).

Дизайн результатів адаптовано для мобільних екранів, забезпечуючи легке читання та навігацію без необхідності зумування або прокрутки.

Размеры Samsung Galaxy A51/71 412 x 914 100% Без ограничения

Режесер:  
Сценарист:  
Вікові обмеження:  
0+

Жанири

- невідомий
- Вистава
- Пригоди
- Анімація
- Дитячий
- Комедія
- Дедиктив
- Документальний
- Драма
- Фентезі
- Фільм-Наур
- Жахи
- Мюзикл
- Тасмниці
- Романтика
- Наукова фантастика
- Трилер
- Війна
- Вестрн

АНАЛІЗУВАТИ

Новини проєкту

Рисунок 3.13 – Інтерфейс для введення даних

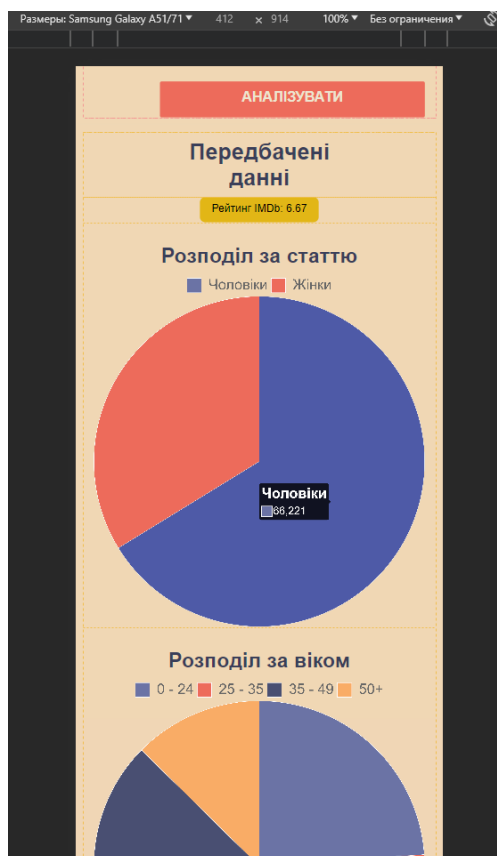


Рисунок 3.14 – Рейтинг IMDb та діаграма розподіл за статтю



Рисунок 3.15 – Діаграма розподілу за віком та професіями

Така структура мобільної адаптації веб-застосунку не лише забезпечує зручність та доступність аналізу мультимедійного контенту для широкого кола користувачів, але й сприяє глибшому залученню аудиторії за рахунок інтуїтивно зрозумілого інтерфейсу та візуально привабливого представлення інформації.

### **Висновки до розділу 3**

Програмна реалізація інтелектуальної системи оцінки контенту є зручним інструментом для технологій аналізу та взаємодії з мультимедійним контентом. Розглянувши ключові компоненти системи, можна зробити наступні висновки:

Архітектура аналітичної системи була спроектована таким чином, щоб забезпечити ефективне збирання, обробку та аналіз мультимедійного контенту. Це включало інтеграцію передових алгоритмів для глибокого розуміння контенту та його контексту, що дозволило системі надавати точні та об'єктивні оцінки.

Застосування нейронних мереж відкрило нові горизонти у точності оцінювання контенту, дозволяючи системі виявляти складні зв'язки та емоційні реакції користувачів на контент. Використання глибокого навчання забезпечило здатність системи адаптуватися та вдосконалюватися з часом.

Створення веб-додатку було важливим етапом у забезпеченні доступності системи для широкого кола користувачів. Веб-додаток став зручним інтерфейсом для взаємодії з системою, пропонуючи легкий доступ до оцінок та аналітики контенту.

Мобільна адаптація стала ключовим аспектом у розширенні досяжності системи, забезпечуючи користувачам можливість доступу до функціоналу системи з будь-якого пристрою. Це підкреслило орієнтацію проекту на сучасні тенденції використання мобільних технологій.

Програмна реалізація інтелектуальної системи оцінки контенту поєднала в собі передові технології та інноваційні підходи, що забезпечили створення надійного, точного та доступного інструменту для оцінки мультимедійного контенту.

## ВИСНОВКИ

У ході виконання даної дипломної роботи було розроблено та впроваджено інтелектуальну систему, здатну автоматизовано оцінювати якість мультимедійного контенту, зокрема фільмів, на основі комплексного аналізу даних про режисерів, сценаристів, жанри та інші характеристики. Система використовує передові методи машинного навчання та алгоритми штучного інтелекту для прогнозування рейтингу фільмів, а також для аналізу розподілу аудиторії за статтю, віком та професією.

Основні результати роботи демонструють значний потенціал використання інтелектуальних систем в аналізі та оцінці мультимедійного контенту. Розроблена система не тільки забезпечує високу точність прогнозування рейтингів фільмів, але й виявляє глибоке розуміння динаміки вподобань різних демографічних груп глядачів. Це відкриває нові можливості для кінопродюсерів та дистриб'юторів у прийнятті обґрунтованих рішень щодо розробки та просування фільмів, з огляду на складні взаємозв'язки між контентом та очікуваннями аудиторії.

Робота також висвітлює важливі виклики та перспективи подальшого розвитку інтелектуальних систем оцінювання контенту. Серед них — необхідність подальшого вдосконалення алгоритмів для забезпечення їх здатності адаптуватися до швидкозмінних трендів у споживанні контенту, а також виклики, пов'язані з обробкою великих обсягів даних у реальному часі.

Зокрема, необхідно забезпечувати прозорість алгоритмів та враховувати потенційний вплив таких систем на різноманітність та доступність контенту.

У підсумку, дана дипломна робота робить вагомий внесок у розвиток інтелектуальних систем для оцінювання мультимедійного контенту, відкриваючи нові напрямки для досліджень та практичного застосування в індустрії розваг. Результати роботи мають значення не тільки для академічної спільноти, але й для професіоналів у сфері медіа, надаючи їм інструменти для більш ефективного управління контентом та взаємодії з аудиторією.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Technical Quality and Perceived Quality in Multimedia Materials. URL: [https://www.researchgate.net/publication/228736426\\_Technical\\_Quality\\_and\\_Perceived\\_Quality\\_in\\_Multimedia\\_Materials-Are\\_They\\_the\\_Same\\_or\\_Different\\_Things](https://www.researchgate.net/publication/228736426_Technical_Quality_and_Perceived_Quality_in_Multimedia_Materials-Are_They_the_Same_or_Different_Things) (дата звернення: 29.01.2024).
2. Content Monetization Explained. URL: <https://www.thinkific.com/blog/how-to-monetize-content/> (дата звернення: 29.01.2024).
3. Return on Investment (ROI). URL: <https://www.investopedia.com/terms/r/returnoninvestment.asp> (дата звернення: 29.01.2024).
4. MEDIA INVESTMENTS. URL: <https://www.oliverwyman.com/our-expertise/insights/2019/apr/media-investments-in-the-digital-era.html> (дата звернення: 29.01.2024).
5. Content Relevance. URL: <https://review.content-science.com/content-relevance-and-usefulness-why-you-need-it-and-4-ways-to-achieve-it> (дата звернення: 29.01.2024).
6. Користувацький досвід хорошого контенту. URL: <https://uxmag.com/articles/the-user-experience-of-good-content> (дата звернення: 29.01.2024).
7. Оцінка мультимедійних завдань. URL: <https://lsa.umich.edu/technology-services/news-events/all-news/teaching-tip-of-the-week/assessing-multimedia-assignments.html> (дата звернення: 29.01.2024).
8. Визначення майбутнього маркетингових досліджень у цифрових та соціальних мережах: перспективи та дослідницькі пропозиції. URL: <https://www.sciencedirect.com/science/article/pii/S0268401220308082> (дата звернення: 29.01.2024).
9. MediaEval Benchmarking Initiative for Multimedia Evaluation. URL: <http://www.multimediaeval.org/> (дата звернення: 29.01.2024).

10. Мультимедійні інструменти у процесах викладання та навчання: систематичний огляд. URL: <https://www.sciencedirect.com/science/article/pii/S2405844020321551> (дата звернення: 29.01.2024).
11. How Web Analytics Helps You Understand User Behavior. URL: <https://www.linkedin.com/pulse/how-web-analytics-helps-you-understand-user-behavior-improve-your/> (дата звернення: 29.01.2024).
12. AI-Powered Personalization. URL: <https://medium.com/@shanukashyap8081/ai-powered-personalization-revolutionizing-digital-marketing-campaigns-8f63addc8ff7> (дата звернення: 29.01.2024).
13. Web analytics. URL: <https://www.hotjar.com/web-analytics/> (дата звернення: 29.01.2024).
14. Netflix Interactive Films. URL: <https://www.linkedin.com/pulse/netflix-interactive-films-pioneering-new-format-sakshi-gupta-cspo--qxphe/> (дата звернення: 29.01.2024).
15. Your Ultimate Guide To Visual Studio vs Visual Studio Code. URL: <https://www.turing.com/kb/ultimate-guide-visual-studio-vs-visual-studio-code> (дата звернення: 29.01.2024).
16. VS Code and Visual Studio Rock the 2022 Stack Overflow Developer Report. URL: <https://visualstudiomagazine.com/articles/2022/06/23/stack-overflow-2022-survey.aspx> (дата звернення: 29.01.2024).
17. Основні можливості та розширення VS Code. URL: <https://code.visualstudio.com/docs/editor/extension-marketplace> (дата звернення: 29.01.2024).
18. What is a VS Code "workspace"? URL: <https://code.visualstudio.com/docs/editor/workspaces> (дата звернення: 29.01.2024).
19. Personalize Visual Studio Code. URL: <https://code.visualstudio.com/docs/introvideos/configure> (дата звернення: 29.01.2024).

20. Version control in VS Code. URL: <https://code.visualstudio.com/docs/introvideos/versioncontrol> (дата звернення: 29.01.2024).

21. FTP/SFTP Extension for Visual Studio Code. URL: <https://artisansweb.net/ftp-sftp-extension-for-visual-studio-code/> (дата звернення: 29.01.2024).

22. What Is Visual Studio Code. URL: <https://webhostingmonkey.com/vs-code/> (дата звернення: 29.01.2024).

23. Telemetry extension authors guide. URL: <https://code.visualstudio.com/api/extension-guides/telemetry> (дата звернення: 29.01.2024).

24. Key Bindings for Visual Studio Code. URL: <https://code.visualstudio.com/docs/getstarted/keybindings> (дата звернення: 29.01.2024).

25. Why Is Python Considered the Top Language? URL: <https://www.bairesdev.com/blog/why-is-python-top-language/> (дата звернення: 29.01.2024).

26. How do hedge fund analysts use Python? URL: <https://www.quora.com/How-do-hedge-fund-analysts-use-Python> (дата звернення: 29.01.2024).

27. Python Microservices: Comprehensive Guide to Building Scalable Architectures. URL: <https://www.temok.com/blog/python-microservices/> (дата звернення: 29.01.2024).

28. NumPy. URL: <https://numpy.org/doc/stable/user/whatisnumpy.html> (дата звернення: 29.01.2024).

29. The N-dimensional array. URL: <https://numpy.org/doc/stable/reference/arrays.ndarray.html> (дата звернення: 29.01.2024).

30. Elevate Math Efficiency: Navigating Numpy Array Operations. URL: <https://www.kdnuggets.com/elevate-math-efficiency-navigating-numpy-array-operations> (дата звернення: 29.01.2024).
31. Reducing NumPy memory usage with lossless compression. URL: <https://pythonspeed.com/articles/numpy-memory-footprint/> (дата звернення: 29.01.2024).
32. Power of NumPy. URL: <https://www.linkedin.com/pulse/power-numpy-fundamental-python-library-numerical-computing-akash-jha-qwgyf/> (дата звернення: 29.01.2024).
33. Numerical operations on arrays. URL: <https://scipy-lectures.org/intro/numpy/operations.html> (дата звернення: 29.01.2024).
34. What is TensorFlow? URL: <https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html> (дата звернення: 29.01.2024).
35. Training speed of TensorFlow. URL: <https://www.neuraldesigner.com/blog/training-speed-comparison-gpu-approximation/> (дата звернення: 29.01.2024).
36. Екосистема TensorFlow. URL: <https://github.com/tensorflow/ecosystem> (дата звернення: 29.01.2024).
37. Deep Dive Into TensorBoard. URL: <https://neptune.ai/blog/tensorboard-tutorial> (дата звернення: 29.01.2024).
38. What is Django? URL: <https://tutorial.djangogirls.org/en/django/> (дата звернення: 29.01.2024).
39. Django ORM and QuerySets. URL: [https://tutorial.djangogirls.org/en/django\\_orm/](https://tutorial.djangogirls.org/en/django_orm/) (дата звернення: 29.01.2024).
40. Movie Dataset. URL: <https://www.kaggle.com/code/mfaaris/content-based-and-recommender-system/input> (дата звернення: 29.01.2024).
41. Багатошаровий перцептрон як інструмент первинної кластеризації зображень / О. В. Гороховатський, О. О. Передрій. (дата звернення: 29.01.2024).



**ДОДАТОК А****Код програми**

Вміст файлу views.py (серверна частина)

```
from django.shortcuts import render
from django.http import JsonResponse
import numpy as np
import json
from . models import *
from django.views import View
from tensorflow import keras

class Index(View):
    def get(self, request):
        list_index = Statia.objects.order_by('-pub_date')
        list_stat = Statia.objects.order_by('-pub_date')
        return render(request, 'movie/index.html', {'list_index':
list_index})
class Detal(View):
    def get(self, request, statia_id):
        return render(request, 'movie/detal.html', {'statia':
Statia.objects.get (id = statia_id )})

class Analyze(View):
    def post(self, request):
        def get_token(text):
            person = Movie.objects.filter(director_ua = text)
            if not person.exists():
                person = Movie.objects.filter(director_en = text)
            if not person.exists():
                return 'text'
            else:
                return person[0].token
        try:
```

```

data = json.loads(json.loads(request.body.decode('utf-
8'))))

director = get_token(data['director'])
writer = get_token(data['writer'])

if writer == 'text' or director == 'text':
    return JsonResponse(status=400)
else:
    x_trein = np.array([[int(data['reit'])/18] + [float(_
for _ in data['categorical'].split()) + [director, writer]])

    dict = {
        'imdb':
round(float(keras.models.load_model('models/imdb_model').predict(x_trein)[0
][0])*10, 2),
        'prof': list(map(lambda x: x*100,
keras.models.load_model('models/prof_model').predict(x_trein)[0].tolist()))
    ,
        'sex': list(map(lambda x: x*100,
keras.models.load_model('models/sex_model').predict(x_trein)[0].tolist())),
        'yer': list(map(lambda x: x*100,
keras.models.load_model('models/yer_model').predict(x_trein)[0].tolist()))),
    }
    return JsonResponse(dict, status=200)
except json.JSONDecodeError:
    return JsonResponse(status=400)
except Exception as e:
    return JsonResponse({'error': str(e)}, status=500)

def get(self, request):
    return JsonResponse({'error': 'Метод GET не відсутній'})

```

Вміст файлу models.py (серверна частина)

```
from django.db import models
```

```
import os

class Movie(models.Model):
    director_en = models.CharField("Прізвище, ім'я en",
max_length=100, unique=True)
    director_ua = models.CharField("Прізвище, ім'я ua",
max_length=100)
    token = models.FloatField()

    def __str__(self):
        return self.director_en

    class Meta:
        verbose_name = 'Режесер'
        verbose_name_plural = 'Режесери'

def get_upload_path(instance, filename):
    return os.path.join('static', 'movie', 'img', filename)

class Statia(models.Model):
    statia_title = models.CharField('Назва статті', max_length = 250)
    html_texst = models.TextField('Код')
    pub_date = models.DateTimeField('Дата публікації')
    model_img = models.ImageField(upload_to = get_upload_path)

    class Meta:
        verbose_name = 'Новина'
        verbose_name_plural = 'Новини'

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

    def __str__(self):
```

```
return self.statia_title
```

JS скрипт (користувацька частина)

```
function drawPieChart(dataValues, dataLabels, id) {  
    var ctx = document.getElementById('myChart_' +  
id).getContext('2d');  
  
    if (window.myCharts && window.myCharts[id]) {  
        window.myCharts[id].destroy();  
    }  
    var myChart = new Chart(ctx, {  
        type: 'pie',  
        data: {  
            labels: dataLabels,  
            datasets: [{  
                data: dataValues,  
                backgroundColor: [  
                    'rgba(107, 115, 165)',  
                    'rgba(237, 107, 91)',  
                    'rgba(73, 79, 114)',  
                    'rgba(249, 172, 102)',  
                    'rgba(144, 190, 109)',  
                    'rgba(204, 129, 153)',  
                    'rgba(108, 166, 205)',  
                    'rgba(202, 166, 115)',  
                    'rgba(80, 130, 158)',  
                    'rgba(214, 134, 107)',  
                    'rgba(159, 189, 97)',  
                    'rgba(103, 128, 159)',  
                    'rgba(196, 97, 104)',  
                    'rgba(104, 162, 143)',  
                    'rgba(217, 147, 105)',  
                    'rgba(136, 154, 198)',  
                ],  
            }],  
        }  
    });  
}
```

```
        borderWidth: 1
      }]
    },
    options: {
      legend: {
        labels: {
          FontSize: 12,
          fontColor: 'black'
        }
      }
    }
  });
  if (!window.myCharts) {
    window.myCharts = {};
  }
  window.myCharts[id] = myChart;
}
```

```
function uploadData() {
  var professions = [
    'адміністратор',
    'митець',
    'лікар',
    'освітянин',
    'інженер',
    'розважальник',
    'керівник',
    'медичний працівник',
    'домогосподар',
    'адвокат',
    'бібліотекар',
    'маркетолог',
    'відсутній',
    'інший',
```

```
'програміст',
'пенсіонер',
'продавець',
'вчений',
'студент',
'технік',
'письменник'
];
var string = '';
for (var i = 1; i < 20; i++) {
    if (document.getElementById('option' + i).checked) {
        string += document.getElementById('option' + i).value + '
';
    } else {
        string += '0 ';
    }
}

var formData = JSON.stringify({
    'director': $('#director_text').val(),
    'writer': $('#text2').val(),
    'reit': $('#single_select').val(),
    'categorical': string
});

const options = {
    method: 'POST',
    headers: {
        'Content-Type': 'application/json',
        "X-CSRFToken": document.querySelector('meta[name="csrf-
token"]').getAttribute('content'),
    },
    body: JSON.stringify(formData),
};
```

```
return new Promise((resolve, reject) => {
  setTimeout(() => {
    reject(new Error('Timeout waiting for response'));
  }, 500);

  fetch('analyze/', options)
    .then(response => {
      if (!response.ok) {
        throw new Error('HTTP error! Status:
${response.status}');
      }
      return response.json();
    })
    .then(data => {
      document.getElementById('imdb').innerHTML = 'Рейтинг
IMDb: ' + data.imdb;
      drawPieChart(data.sex, ['Чоловіки', 'Жінки'], 'sex');
      drawPieChart(data.yer, ['0 - 24', '25 - 35', '35 -
49', '50+'], 'yer');
      drawPieChart(data.prof, professions, 'prof');
      document.getElementById("result").style.display =
"flex";
      document.getElementById("error").style.display =
"none";
      resolve(data);
    })
    .catch(error => {
      document.getElementById("result").style.display =
"none";
      document.getElementById("error").style.display =
"flex";
      document.getElementById('error').innerHTML = 'Режесера
або сценариста у базі не знайдено';
    });
});
```

```
        reject(error);
    });
});
}
```

Код створення моделей (не є частиною проєкту)

```
from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.metrics import MeanAbsolutePercentageError
import numpy as np
import matplotlib.pyplot as plt
import openpyxl
import os

from f import loaddata

import tensorflow as tf
tf.compat.v1.logging.set_verbosity(tf.compat.v1.logging.ERROR)

X_train_text, X_train_numeric, y_train = loaddata()

# Токенизація текстових даних
tokenizer = Tokenizer()
tokenizer.fit_on_texts(X_train_text)
X_train_seq = tokenizer.texts_to_sequences(X_train_text)
max_seq_length = max(len(seq) for seq in X_train_seq)
X_train_pad = pad_sequences(X_train_seq, maxlen=max_seq_length,
padding='post')
min_token = min([item for sublist in X_train_pad.tolist() for item in
sublist])
```



```
max_token = max([item for sublist in X_train_pad.tolist() for item in
sublist])
norm_token_list = [[(token - min_token) / (max_token - min_token) for
token in seq] for seq in X_train_pad.tolist()]
x_trein = np.array([X_train_numeric[i] + norm_token_list[i] for i in
range(len(norm_token_list))])
y_trein = np.array(y_trein)
```

```
learning_rate = 0.00001
```

```
model = Sequential([
    Dense(16, input_dim=22, activation='relu'),
    Dense(8, activation='relu'),
    Dense(4, activation='softmax'),
])
...
```

```
model = Sequential([
    Dense(10, input_dim=22, activation='relu'),
    Dense(1, activation='sigmoid')
])
```

```
model.compile(optimizer = 'adam' ,
              loss = 'mean_absolute_error',
              metrics = [MeanAbsolutePercentageError()])
...
```

```
model.compile(optimizer = 'adam' ,
              loss = 'categorical_crossentropy',
              metrics = ['accuracy'])
```

```
history = model.fit(x_trein, y_trein, epochs=10, batch_size=1,
verbose=1, validation_split = 0.2)

model.summary()

y_pred = model.predict(x_trein)
for i in range(int(len(x_trein)/25)):
    print(y_trein[i])
    print(y_pred[i])
    print('-----')

wb = openpyxl.Workbook()
sheet = wb.create_sheet('Лист1')
# Получение значений потерь и точности из истории обучения
...

loss = history.history['loss']
val_loss = history.history['val_loss']
accuracy = history.history['accuracy']
val_accuracy = history.history['val_accuracy']
...

loss = history.history['loss']
val_loss = history.history['val_loss']
accuracy = history.history['accuracy']
val_accuracy = history.history['val_accuracy']
...

accuracy = history.history['mean_absolute_percentage_error']
val_accuracy = history.history['val_mean_absolute_percentage_error']
...

for i in range(len(loss)):
    sheet[f'A{i+1}'] = i+1
    sheet[f'B{i+1}'] = loss[i]
    sheet[f'C{i+1}'] = val_loss[i]
```

```
sheet[f'D{i+1}'] = accuracy[i]
sheet[f'E{i+1}'] = val_accuracy[i]
...

sheet[f'D{i+1}'] = (100 - accuracy[i])/100
sheet[f'E{i+1}'] = (100 - val_accuracy[i])/100
...

wb.save('token.xlsx')
model_directory = '2yer_model'
if not os.path.exists(model_directory):
    os.makedirs(model_directory)

model.save(model_directory)
```