Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: Розробка системи для багатофакторної класифікації фільмів

Виконав(ла): студент(ка) 4 курсу, групи СН-42
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

(підпис) Новіцький М.Р. (прізвище та ініціали)

Керівник (підпис) Бревус Г.Б. (прізвище та ініціали)

Нормоконтроль (підпис) Липак Г.І. (прізвище та ініціали)

Завідувач кафедри (підпис) Боднарчук І.О. (прізвище та ініціали)

Рецензент (підпис) (прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
"27" січня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

студенту Новіцький Максим Русланович
(прізвище, ім'я, по батькові)

1. Тема роботи **Розробка системи для багатофакторної класифікації фільмів.**

Керівник роботи **асист. Бревус Г.Б.**
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від "*07*" *травня* 2025 року № *4/7-444*

2. Термін подання студентом завершеної роботи 20 червня 2025 р.

3. Вихідні дані до роботи Літературні джерела з тематики роботи

4. Зміст роботи (перелік питань, які потрібно розробити)
ВСТУП 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ 1.1 Методи класифікації з багатьма мітками (багатофакторної класифікації) 1.2 Методи трансформації задачі (Problem Transformation – PT) 1.3 Методи адаптації алгоритмів 2 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ БАГАТОФАКТОНОЇ КЛАСИФІКАЦІЇ 2.1 Постановка задачі прикладу 2.2 Опис набору даних для прикладу 2.3 Опис програмного рішення 3 РЕАЛІЗАЦІЯ ЗАДАЧІ БАГАТОФАКТОРНОЇ КЛАСИФІКАЦІЇ ФІЛЬМІВ 3.1 Опис задачі багатофакторної класифікації фільмів 3.2 Налаштування задачі класифікації з кількома мітками 3.3 Про набір даних 3.4 Стратегія побудови моделі прогнозування жанрів фільмів 3.5 Програмна реалізація 3.6 Перетворення тексту на функції 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ; ВИСНОВКИ; СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів):
Тема. Мета та актуальність роботи. Види класифікації. Властивості багатофакторної класифікації. Сфери застосування. Складнощі багатоміткової класифікації. Підходи до побудови моделі. Постановка задачі. Вхідні дані. Попередня обробка даних. Перетворення цільових змінних. Побудова моделі навчання. Оцінка моделі. Перетворення міток у текст. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Окіпний І.Б., к.т.н., доцент кафедри МТ		

7. Дата видачі завдання 27 січня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	27.01.2025 – 31.01.2025	*Виконано*
2.	Підбір джерел по темі роботи	01.02.2025 – 01.04.2025	*Виконано*
3.	Оформлення першого розділу	15.04.2025	*Виконано*
4.	Оформлення другого розділу	20.04.2025	
5.	Оформлення третього розділу	30.04.2025	*Виконано*
6.	Виконання завдання до підрозділу "Безпека		
7.	життєдіяльності, основи охорони праці"	15.05.2025	*Виконано*
8.	Оформлення кваліфікаційної роботи	07.06.2025	*Виконано*
9.	Перевірка на плагіат	07.06.2025	*Виконано*
10.	Нормоконтроль	09.06.2025	*Виконано*
11.	Попередній захист кваліфікаційної роботи	11.06.2025	*Виконано*
12.	Захист кваліфікаційної роботи	23.06.2025	
13.			
14.			
15.			

Студент _____ Новіцький М.Р.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Бревус Г.Б.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

"Розробка системи для багатофакторної класифікації фільмів." // Кваліфікаційна робота освітнього рівня "Бакалавр" // Новіцький Максим Русланович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-42 // Тернопіль, 2025 // с. – 56, рис. – 18, таблиць – 6, джерел – 15.

Ключові слова: машинне навчання, Python, багатофакторна класифікація, алгоритм, набір даних

Багатофакторна класифікація (multilabel classification) є одним із підтипів задач машинного навчання, де кожному об'єкту може відповідати не один, а кілька класів одночасно. Це суттєво відрізняється від традиційної однофакторної класифікації, де кожен приклад має лише одну мітку. Multilabel підхід широко застосовується у багатьох сферах, зокрема в обробці текстів, медичних діагнозах, розпізнаванні музики, а також у класифікації фільмів за жанрами.

У задачах класифікації фільмів багатофакторність обумовлена тим, що більшість сучасних кінострічок не обмежується одним жанром. Наприклад, один і той же фільм може бути класифікований як драма, трилер і кримінальний. Завдання системи машинного навчання полягає у правильному визначенні набору жанрів для кожного фільму на основі доступної інформації, зокрема опису сюжету, акторського складу, ключових слів, а іноді й текстових рецензій.

Одним із викликів у такій класифікації є незбалансованість класів – деякі жанри значно частіше зустрічаються, ніж інші, що ускладнює навчання. Ще одна проблема – висока кореляція між жанрами, яку не завжди легко врахувати при використанні незалежних моделей. Крім того, жанри мають нечіткі межі, а їх трактування часто суб'єктивне.

Таким чином, багатофакторна класифікація є важливим напрямком у сфері машинного навчання з великою практичною цінністю. Задачі класифікації фільмів є одними з найяскравіших прикладів її застосування, оскільки поєднують роботу з текстовими, семантичними та контекстуальними даними для створення точних багатожанрових передбачень.

ANNOTATION

"Development of a System for Multi-Factor Movie Classification" // Qualification work of the educational level "Bachelor" // Novitskyi Maksym // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, Group CH-42 // Ternopil, 2025 // p. – 56, fig. – 18, tables – 6, references – 15.

Keywords: machine learning, Python, multilabel classification, algorithm, dataset

Multilabel classification is a subtype of machine learning tasks where each object can belong to multiple classes simultaneously. This fundamentally differs from traditional single-label classification, in which each instance is assigned only one label. The multilabel approach is widely used across various fields, including text processing, medical diagnosis, music recognition, and particularly in film genre classification.

In movie classification tasks, multilabeling arises from the fact that most modern films do not belong to a single genre. For example, a film may simultaneously be classified as a drama, thriller, and crime story. The goal of a machine learning system is to accurately determine the set of genres for each movie based on the available information, including the plot description, cast, keywords, and sometimes even textual reviews.

One of the challenges in such classification tasks is class imbalance—some genres occur much more frequently than others, making it harder to train reliable models. Another issue is the high correlation between genres, which is not always easy to capture using independent models. Moreover, genre boundaries are often fuzzy, and their interpretation can be subjective.

Therefore, multilabel classification is a significant area within machine learning with substantial practical value. Movie genre classification tasks are among the most

illustrative examples of its application, as they combine the processing of textual, semantic, and contextual data to generate accurate multi-genre predictions.

ЗМІСТ

ВСТУП

Одна з найбільш використовуваних можливостей методів машинного навчання з учителем – це класифікація, яка застосовується в багатьох контекстах, таких як визначення позитивного чи негативного відгуку про ресторан, або визначення наявності на зображенні кота чи собаки. Це завдання можна розділити на три області: бінарна класифікація, багатокласова класифікація та багатофакторна класифікація. У цій роботі ми пояснимо ці типи класифікації та розглянемо, чому вони відрізняються один від одного, а також покажемо реальний сценарій, де можна використовувати багатофакторну класифікацію.

Бінарна класифікація використовується, коли є лише два різних класи, і дані, які ми хочемо класифікувати, належать виключно до одного з цих класів, наприклад, щоб класифікувати публікацію про певний продукт як позитивну чи негативну.

Багатокласова класифікація використовується, коли є три або більше класів, і дані, які ми хочемо класифікувати, належать виключно до одного з цих класів, наприклад, щоб класифікувати, чи світлофор на зображенні червоний, жовтий чи зелений.

Класифікація за кількома мітками (багатофакторна) використовується, коли є два або більше класів, і дані, які ми хочемо класифікувати, можуть не належати до жодного з класів або до всіх одночасно, наприклад, для класифікації дорожніх знаків, що містяться на зображенні.

Сьогодні сучасні методи класифікації з кількома мітками все частіше потребують таких застосувань, як класифікація в органічній хімії, категоризація музики та класифікація семантичних сцен. Вона також вносить свій доробок у визначення концепцій для кількісної оцінки багатоміткової природи набору даних.

Традиційна класифікація за однією міткою пов'язана з навчанням на наборі прикладів, пов'язаних з однією міткою l з набору окремих міток L, $|L| > 1$. Якщо

|L| = 2, то задача навчання називається бінарною задачею класифікації (або фільтрацією у випадку текстових та веб-даних), тоді як якщо |L| > 2, то вона називається багатокласовою задачею класифікації.

У багатомітковій класифікації приклади пов'язані з набором міток $Y \subseteq L$. У минулому багатоміткова класифікація була головним чином мотивована завданнями категоризації тексту та медичної діагностики. Текстові документи зазвичай належать до кількох концептуальних класів. Наприклад, газетну статтю про реакцію християнської церкви на вихід фільму «Код да Вінчі» можна класифікувати за обома категоріями: Суспільство\Релігія та Мистецтво\Фільми. Аналогічно, у медичній діагностиці пацієнт може страждати, наприклад, від діабету та анемії одночасно.

Сьогодні ми помічаємо, що сучасні застосування все частіше вимагають багатоміткових методів класифікації, таких як класифікація в хімії білків [2], в економіці [1], категоризація музики [9] та семантична класифікація сцен [4]. У семантичній класифікації сцен фотографія може належати до кількох концептуальних класів, таких як захід сонця та пляжі одночасно. Аналогічно, в категоризації музики пісня може належати до кількох жанрів. Наприклад, багато хітів популярних рок-гуртів можна охарактеризувати як рок, так і баладу.

Ця робота має за мету слугувати відправною точкою та довідником для дослідників, які цікавляться багатомітковою (багатофакторною) класифікацією.

Основні задачі в роботі:

а) структурований аналіз літератури про методи багатоміткової класифікації з висновками щодо їхніх відносних сильних та слабких сторін, а також, по можливості, абстрагування конкретних методів до більш загальних і, отже, корисніших схем;

б) введення недокументованого багатоміткового методу;

в) визначення концепції кількісної оцінки багатоміткової природи набору даних;

г) попередні порівняльні експериментальні результати щодо ефективності певних багатоміткових методів.

1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

Завдання, яке також належить до загальної групи навчання з учителем і дуже актуальне для багатоміткової класифікації, – це ранжування. При ранжуванні завдання полягає в тому, щоб упорядкувати набір міток L так, щоб найвищі мітки були більше пов'язані з новим екземпляром. Існує ряд методів багатоміткової класифікації, які вивчають функцію ранжування з багатоміткових даних. Однак ранжування міток вимагає постобробки, щоб отримати набір міток, який є належним результатом багатоміткового класифікатора.

У певних задачах класифікації мітки належать до ієрархічної структури. Наприклад, відкритий каталог деякі пошукові системи і каталоги онлайн-ресурсів підтримують ієрархію концептуальних класів для категоризації веб-сторінок. Веб-сторінка може бути позначена за допомогою одного або кількох із цих класів, які можуть належати до різних рівнів ієрархії.

Верхній рівень ієрархії MIPS (Мюнхенський інформаційний центр білкових послідовностей) (http://mips.gsf.de/) складається з таких класів, як: метаболізм, енергія, транскрипція та синтез білка. Кожен із цих класів потім поділяється на більш специфічні класи, які, у свою чергу, поділяються, а потім знову поділяються, таким чином ієрархія має глибину до 4 рівнів [6]. Коли мітки в наборі даних належать до ієрархічної структури, то завдання називається ієрархічною класифікацією. Якщо кожен приклад позначено більш ніж одним вузлом ієрархічної структури, то завдання називається ієрархічною багатомітковою класифікацією. У цій роботі ми зосередимося на плоских (неієрархічних) методах класифікації з кількома мітками.

Автори [8] називають задачі з кількома мітками задачами напівконтрольованої класифікації, де кожен приклад пов'язаний з кількома класами, але лише один з цих класів є справжнім класом прикладу. Це завдання не таке поширене в реальних застосуваннях, як те, яке ми вивчаємо.

Багатоетапне навчання – це різновид навчання з учителем, де завдання полягає у вивченні концепції на основі позитивних та негативних пакетів екземплярів [**Error! Reference source not found.**]. Кожен пакет може містити б агато екземплярів, але пакет позначається як позитивний, навіть якщо лише один з екземплярів у ньому відповідає концепції. Пакет позначається як негативний, лише якщо всі екземпляри в ньому є негативними.

1.1 Методи класифікації з багатьма мітками (багатофакторної класифікації)

Ми можемо розділити існуючі методи класифікації за кількома мітками на дві основні категорії:

а) методи перетворення задачі та

б) методи адаптації алгоритмів.

Методами перетворення задачі ми називаємо ті методи, які перетворюють задачу класифікації за кількома мітками на одну або декілька задач класифікації або регресії з однією міткою, для обох з яких існує велика кількість алгоритмів навчання. Методами адаптації алгоритмів ми називаємо ті методи, які розширюють специфічні алгоритми навчання для безпосередньої обробки даних з кількома мітками.

1.2 Методи трансформації задачі (Problem Transformation – PT)

Для ілюстрації цих методів ми використаємо набір даних таблиці 1.1. Він складається з чотирьох прикладів (у цьому випадку документів), що належать до одного або кількох із чотирьох класів: Спорт, Релігія, Наука та Політика.

Існують два прості методи перетворення задачі, які примушують навчальну задачу до традиційної класифікації за однією міткою [4]. Перший метод (який отримав назву PT1) суб'єктивно або випадково вибирає одну з кількох міток кожного екземпляра з кількома мітками та відкидає решту, тоді як

другий метод (який отримав назву PT2) просто відкидає кожен екземпляр з кількома мітками з набору даних з кількома мітками.

Таблиця 1.1 – Приклад набору даних з кількома мітками

№	Спорт	Релігія	Наука	Політика
1	X			X
2			X	X
3	X			
4		X	X	

У таблиці 1.2 та таблиці 1.3 показано перетворений набір даних за допомогою методів PT1 та PT2 відповідно. Ці два методи перетворення задачі відкидають значну частину інформаційного вмісту вихідного набору даних з кількома мітками і тому далі не розглядаються в цій роботі.

Таблиця 1.2 – Трансформований набір даних за допомогою PT1

№	Спорт	Релігія	Наука	Політика
1	X			
2				X
3	X			
4		X		

Таблиця 1.3 – Трансформований набір даних за допомогою PT2

№	Спорт	Релігія	Наука	Політика
3	X			

Третій метод перетворення задачі, який ми згадаємо (під назвою PT3), розглядає кожен різний набір міток, що існують у наборі даних з кількома мітками, як одну мітку. Таким чином, він вивчає один класифікатор з однією

міткою H: X → P(L), де P(L) – степенева множина L. У таблиці 1.4 показано результат перетворення набору даних з таблиці 1 за допомогою цього методу. Одним із негативних аспектів PT3 є те, що він може призвести до наборів даних з великою кількістю класів та невеликою кількістю прикладів на клас. PT3 використовувався в минулому в, наприклад, [4, 7].

Таблиця 1.4 – Трансформований набір даних за допомогою PT3

№	Спорт	(Спорт ∧ Політика)	(Наука ∧ Політика)	(Наука ∧ Релігія)
1		X		
2			X	
3	X			
4				X

Найпоширеніший метод перетворення задачі (який отримав назву PT4) вивчає | L | бінарні класифікатори $H_l: X \to \{ l , \neg l \}$, по одному для кожної різної мітки l в L. Він перетворює вихідний набір даних на | L | набори даних D_l, які містять усі приклади вихідного набору даних, позначені як l , якщо мітки вихідного прикладу містили l, та як ¬l в іншому випадку. Це те саме рішення, яке використовується для вирішення задачі з однією міткою та кількома класами за допомогою бінарного класифікатора.

Для класифікації нового екземпляра x цей метод виводить як набір міток об'єднання міток, що виводяться класифікаторами |L| :

$$H_{PT4}(x) = \bigcup_{l \in L} \{l\} : H_l(x) = l \tag{1.1}$$

На рисунку 1.1 показано чотири набори даних, побудовані за допомогою PT4 при застосуванні до набору даних таблиці 1.1.

№	Спорт	¬Спорт
1	X	
2		X
3	X	
4		X

а)

№	Політика	¬Політика
1	X	
2	X	
3		X
4		X

б)

№	Релігія	¬Релігія
1	X	
2	X	
3		X
4		X

в)

№	Наука	¬Наука
1		X
2	X	
3		X
4	X	

г)

Рисунок 1.1 – Чотири набори даних, побудовані за допомогою PT4

Простий, але недокументований метод перетворення задачі є наступним (отримав назву PT5).

По-перше, він розкладає кожен приклад (x , Y) на $\left| Y \right|$ екземпляри (x , l) для всіх $l \in Y$. Потім він вивчає один класифікатор на основі покриття з однією міткою з перетвореного набору даних. Класифікатори розподілу – це ті класифікатори, які можуть виводити розподіл ступенів достовірності (або ймовірностей) для всіх міток у L. Нарешті, він обробляє цей розподіл, щоб вивести набір міток. Один простий спосіб досягти цього – вивести ті мітки, для яких ступінь достовірності перевищує певний поріг (наприклад, 0,5). Більш складний спосіб – вивести ті мітки, для яких ступінь достовірності перевищує відсоток (наприклад, 70%) від найвищого ступеня достовірності. У таблиці 1.5 показано результат перетворення набору даних з таблиці 1.1 з використанням цього методу.

1.3 Методи адаптації алгоритмів

В роботі [6] автори адаптували алгоритм для даних з кількома мітками. Вони модифікували формулу розрахунку ентропії наступним чином:

$$entropy(S) = -\sum_{i=1}^{N} (p(c_i)\log p(c_i) + q(c_i)\log q(c_i)), \qquad (1.2)$$

де $p(c_i)$ = відносна частота класу c_i та $q(c_i) = 1 - p(c_i)$. Вони також дозволили кілька міток у листках дерева.

Таблиця 1.5 – Трансформований набір даних за допомогою PT5

№	Клас
1	Спорт
1	Політика
2	Наука
2	Політика
3	Спорт
4	Релігія
4	Наука

Adaboost.MH та Adaboost.MR – це два розширення AdaBoost для класифікації з кількома мітками [10]. Обидва застосовують AdaBoost до слабких класифікаторів виду $H : X \times L \rightarrow R$. В AdaBoost.MH, якщо знак виводу слабких класифікаторів додатний для нового прикладу x та мітки l, то ми вважаємо, що цей приклад можна позначити l, тоді як якщо він від'ємний, то цей приклад не позначається l. В AdaBoost.MR вивід слабких класифікаторів враховується для ранжування кожної з міток у L.

Хоча ці два алгоритми є адаптаціями певного підходу до навчання, ми помічаємо, що в своїй основі вони фактично використовують перетворення

задачі (отримане назвою PT6): кожен екземпляр (x, Y) розкладається на $|L|$ екземплярів (x, l, $Y[l]$) для всіх $l \in L$, де $Y[l] = 1$, якщо $l \in Y$, а $[l] = -1$ в іншому випадку. У таблиці 1.6 показано результат перетворення набору даних з таблиці 1.1 за допомогою цього методу.

Таблиця 1.6 – Трансформований набір даних за допомогою PT6

Наприклад	L	$Y[l]$
1	Спорт	1
1	Релігія	-1
1	Наука	-1
1	Політика	1
2	Спорт	-1
2	Релігія	-1
2	Наука	1
2	Політика	1
3	Спорт	1
3	Релігія	-1
3	Наука	-1
3	Політика	-1
4	Спорт	-1
4	Релігія	1
4	Наука	1
4	Політика	-1

ML- *k*NN [6**Error! Reference source not found.**] – це адаптація алгоритму л інивого навчання *k*NN для даних з кількома мітками. Фактично, цей метод відповідає парадигмі PT4. По суті, ML- *k*NN використовує алгоритм *k*NN незалежно для кожної мітки l: він знаходить k найближчих прикладів до тестового екземпляра та розглядає ті, що позначені принаймні l, як позитивні, а решту – як негативні. Основна відмінність цього методу від застосування оригінального алгоритму *k*NN до перетвореної задачі з використанням PT4 полягає у використанні апріорних ймовірностей. ML- *k*NN також має можливість генерувати ранжування міток як вихідний результат.

Луо та Зінкір -Хейвуд (2005) представляють дві системи для класифікації документів з кількома мітками, які також базуються на класифікаторі *k* нейронних мереж. Основний внесок їхньої роботи полягає на етапі попередньої обробки для ефективного представлення документів. Для класифікації нового екземпляра системи спочатку знаходять *k* найближчих прикладів. Потім для кожної появи кожної мітки в кожному з цих прикладів вони збільшують відповідний лічильник для цієї мітки. Нарешті, вони виводять *N* міток з найбільшою кількістю. *N* вибирається на основі кількості міток екземпляра. Це невідповідна стратегія для реального використання, де кількість міток нового екземпляра невідома.

MMAC [8] – це алгоритм, що відповідає парадигмі асоціативної класифікації, яка має справу з побудовою наборів правил класифікації за допомогою асоціативного аналізу правил. MMAC вивчає початковий набір правил класифікації за допомогою асоціативного аналізу правил, видаляє приклади, пов'язані з цим набором правил, і рекурсивно вивчає новий набір правил з решти прикладів, доки не залишиться більше частий елемент. Ці кілька наборів правил можуть містити правила з подібними передумовами, але різними мітками праворуч. Такі правила об'єднуються в одне правило з кількома мітками. Мітки ранжуються відповідно до підтримки відповідних окремих правил.

2 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ БАГАТОФАКТОНОЇ КЛАСИФІКАЦІЇ

2.1 Постановка задачі прикладу

Цей розділ містить покроковий аналіз створення ефективної моделі машинного навчання для класифікації тексту з кількома мітками. Ми використовуватимемо DeBERTa як базову модель, яка наразі є найкращим вибором для моделей кодувальників, і налаштуємо її на нашому наборі даних. Цей набір даних містить 3140 ретельно перевірених навчальних прикладів значущих бізнес-подій у біотехнологічній галузі. Хоча тема є специфічною, набір даних є універсальним і надзвичайно корисним для різних завдань класифікації бізнес-даних, тому використані алгоритми підійдуть для багатофакторної класифікації фільмів.

Класифікація тексту є найпоширенішим завданням NLP. Незважаючи на просте формулювання, для більшості реальних бізнес-випадків це складне завдання, яке вимагає експертних знань для збору високоякісних наборів даних та одночасного навчання продуктивних і точних моделей. Класифікація за кількома мітками є ще складнішою та проблематичнішою. Ця галузь досліджень недостатньо представлена в спільноті машинного навчання, оскільки публічні набори даних занадто прості для навчання практичних моделей. Більше того, інженери часто порівнюють моделі аналізу настроїв та інші відносно прості завдання, які не відображають реальної здатності моделей вирішувати проблеми. Ми навчимо модель класифікації за кількома мітками на наборі даних з відкритим кодом, щоб довести, що кожен може розробити краще рішення.

Перш ніж розпочати проект, переконайтеся, що ви встановили такі пакети:

```
!pip install datasets transformers evaluate sentencepiece accelerate
```

1. Datasets: це потужний інструмент, який забезпечує єдиний та простий у використанні інтерфейс для доступу та роботи з широким спектром наборів даних, що зазвичай використовуються в дослідженнях та застосуваннях NLP.

2. Transformers: бібліотека для роботи з попередньо навченими моделями в обробці природної мови, яка надає велику колекцію сучасних моделей для таких завдань, як генерація тексту, переклад, аналіз настроїв тощо.

3. Evaluate: бібліотека для легкої оцінки моделей машинного навчання та наборів даних.

4. Sentencepiece: це токенізатор та детокенізатор тексту, який використовується переважно для систем генерації тексту на основі нейронних мереж, де розмір словникового запасу визначається заздалегідь до навчання нейронної моделі.

5. Accelerate: бібліотека, що забезпечує високорівневий інтерфейс для розподіленого навчання, що спрощує для дослідників та практиків масштабування робочих процесів глибокого навчання на кількох графічних процесорах або машинах.

2.2 Опис набору даних для прикладу

У відкритому доступі наявний набір даних [5], спеціально розроблений для сектору біотехнологічних новин, прагнучи подолати обмеження існуючих бенчмарків. Цей набір даних багатий на складний контент, що містить тисячі статей з біотехнологічних новин, що охоплюють різні важливі бізнес-події, тим самим забезпечуючи більш детальне уявлення про проблеми вилучення інформації.

Набір даних охоплює 31 клас, включаючи категорію «Немає», для охоплення різних подій та типів інформації, таких як організація подій, заяви керівництва, схвалення регуляторних органів, оголошення про найм тощо.

Ключові аспекти набору даних:

- Вилучення подій.
- Багатокласова класифікація.
- Домен новин біотехнологій.
- 31 клас.
- Загальна кількість прикладів: 3140.

Мітки набору даних:

- event organization – організація або участь у заході, такому як конференція, виставка тощо.
- executive statement – заява або цитата керівника компанії.
- regulatory approval – отримання схвалення від регуляторних органів на продукти, послуги, випробування тощо.
- hiring – оголошення про нові прийоми на роботу або призначення в компанії.
- foundation – створення нового благодійного фонду.
- closing – закриття об’єкта/офісу/підрозділу або припинення ініціативи.
- partnerships & alliances – формування партнерств або стратегічних альянсів з іншими компаніями.
- expanding industry – розширення на нові галузі чи ринки.
- new initiatives or programs – оголошення нових ініціатив, програм чи кампаній.
- m&a – злиття, поглинання або продаж активів і ще 21 інші.

2.3 Опис програмного рішення

Перш за все, ініціалізуємо набір даних та виконаємо попередню обробку класів (лістинг 2.1).

Лістинг 2.1 – Попередня обробка набору даних

```
from datasets import load_dataset

dataset = load_dataset('knowledgator/events_classification_biotech')

classes = [class_ for class_ in dataset['train'].features['label 1'].names if class_]
class2id = {class_:id for id, class_ in enumerate(classes)}
id2class = {id:class_ for class_, id in class2id.items()}
```

Після цього ми токенізуємо набір даних та обробляємо мітки для класифікації за кількома мітками. По-перше, ми збираємося ініціалізувати токенаізатор . У нашому випадку ми використовуватимемо модель DeBERTa, яка наразі є найкращим вибором для моделей на основі кодування (лістинг 2.2).

Лістинг 2.2 – Підключення токенайзеру

```
from transformers import AutoTokenizer

model_path = 'microsoft/deberta-v3-small'

tokenizer = AutoTokenizer.from_pretrained(model_path)
```

Потім ми токенізуємо набір даних та обробляємо мітки для класифікації з кількома мітками (лістинг 2.3).

Лістинг 2.3 – Присвоєння міток елементам даних

```
def preprocess_function(example):
    text = f"{example['title']}.\n{example['content']}"
    all_labels = example['all_labels'].split(', ')
    labels = [0. for i in range(len(classes))]
    for label in all_labels:
        label_id = class2id[label]
        labels[label_id] = 1.

    example = tokenizer(text, truncation=True)
    example['labels'] = labels
    return example

tokenized_dataset = dataset.map(preprocess_function)
```

Після цього ми ініціалізуємо DataCollatorWithPadding. Ефективніше динамічно доповнювати речення до максимальної довжини в пакеті під час сортування, ніж доповнювати весь набір даних до максимальної довжини.

Лістинг 2.4 – ініціалізація DataCollatorWithPadding

```
from transformers import DataCollatorWithPadding

data_collator = DataCollatorWithPadding(tokenizer=tokenizer)
```

Впровадження метрик під час навчання надзвичайно корисне для моніторингу продуктивності моделі за часом. Це може допомогти уникнути надмірного налаштування та побудувати більш загальну модель.

Лістинг 2.5 – Використання метрик для моделі машинного навчання

```
import evaluate
import numpy as np

clf_metrics = evaluate.combine(["accuracy", "f1", "precision", "recall"])

def sigmoid(x):
   return 1/(1 + np.exp(-x))

def compute_metrics(eval_pred):

   predictions, labels = eval_pred
   predictions = sigmoid(predictions)
   predictions = (predictions > 0.5).astype(int).reshape(-1)
   return clf_metrics.compute(predictions=predictions, references=labels.astype(int)
references=labels.astype(int).reshape(-1))
```

Тепер ініціалізуємо нашу модель та передамо всі необхідні деталі про наше завдання класифікації, такі як кількість міток, назви класів та їх ідентифікатори, а також тип класифікації.

Лістинг 2.6 – Передача параметрів моделі класифікації

```
from transformers import AutoModelForSequenceClassification, TrainingArguments, Trainer

model = AutoModelForSequenceClassification.from_pretrained(

    model_path, num_labels=len(classes),
            id2label=id2class, label2id=class2id,
                        problem_type = "multi_label_classification"
```

Далі нам потрібно налаштувати аргументи навчання, а потім ми можемо розпочати процес навчання.

Лістинг 2.7 – Налаштування параметрів моделі класифікації

```
training_args = TrainingArguments(

    output_dir="my_awesome_model",
    learning_rate=2e-5,
    per_device_train_batch_size=3,
    per_device_eval_batch_size=3,
    num_train_epochs=2,
    weight_decay=0.01,
    evaluation_strategy="epoch",
    save_strategy="epoch",
    load_best_model_at_end=True,
)

trainer = Trainer(

    model=model,
    args=training_args,
    train_dataset=tokenized_dataset["train"],
    eval_dataset=tokenized_dataset["test"],
    tokenizer=tokenizer,
    data_collator=data_collator,
    compute_metrics=compute_metrics,
)

trainer.train()
```

Процедура навчання проста та дає вражаючі результати. Ми створили її, використовуючи виключно рішення з відкритим кодом, і досягли чудових результатів.

У цьому розділі ми продемонстрували, як створити багатомітковий класифікатор тексту з нуля за допомогою Transformers and Datasets. Ми використали модель кодувальника DeBERTa та налаштували її на наборі даних біотехнологічних новин з 31 міткою. Після токенізації тексту та кодування багатофункціональних міток ми налаштували цикл навчання з корисними метриками для моніторингу. Модель тепер може точно класифікувати статті біотехнологічних новин за кількома релевантними категоріями. Цей комплексний робочий процес демонструє, як нові набори даних можна підготувати та ввести в потужні попередньо навчені моделі, такі як DeBERTa, для створення власних та потужних класифікаторів тексту для реальних застосувань.

3 РЕАЛІЗАЦІЯ ЗАДАЧІ БАГАТОФАКТОРНОЇ КЛАСИФІКАЦІЇ ФІЛЬМІВ

3.1 Опис задачі багатофакторної класифікації фільмів

Класифікація з кількома мітками виникла в результаті дослідження проблеми категоризації тексту, де кожен документ може належати до кількох попередньо визначених тем одночасно.

Класифікація текстових даних за багатьма мітками є важливою проблемою. Приклади варіюються від статей новин до електронних листів. Наприклад, це можна використовувати, щоб знайти жанри, до яких належить фільм, на основі короткого змісту його сюжету (рисунок 3.1).

Рисунок 3.1 – Класифікація за кількома мітками для пошуку жанрів на основі кінопостерів

У класифікації з кількома мітками навчальний набір складається з екземплярів, кожен з яких пов’язаний із набором міток, і завдання полягає в тому,

щоб передбачити набори міток невидимих екземплярів шляхом аналізу навчальних екземплярів із відомими наборами міток.

Різниця між класифікацією з кількома класами та класифікацією з кількома мітками полягає в тому, що в задачах із кількома класами класи є взаємовиключними, тоді як для задач із кількома мітками кожна мітка представляє інше завдання класифікації, але завдання якимось чином пов’язані.

Наприклад, багатокласова класифікація передбачає, що кожному зразку присвоєно одну й лише одну мітку: фрукт може бути або яблуком, або грушею, але не обома одночасно. Тоді як прикладом класифікації з кількома мітками може бути те, що текст може стосуватися будь-якої релігії, політики, фінансів чи освіти одночасно або нічого з цього.

Класифікація фільмів, наприклад, – це задача класифікації тексту з кількома мітками з дуже незбалансованим набором даних.

Перед нами стоїть завдання побудувати модель із кількома мітками, здатну виявляти різні типи фільмів, як-от погрози, непристойність, образи та ненависть на основі особи. Нам потрібно створити модель, яка передбачає ймовірність кожного типу токсичності для кожного коментаря.

У цьому розділі демонструються методи штучного інтелекту для прогнозування жанру фільму. Тож очевидною задачею є знайти спосіб перетворити це формулювання задачі на текстові дані. Зараз більшість навчальних посібників з NLP розглядають вирішення задач класифікації з однією міткою (коли є лише одна мітка на спостереження).

Але фільми не одновимірні. Один фільм може охоплювати кілька жанрів. Саме цей виклик реалізується в цій роботі. Спочатку ми отримали набір даних у вигляді опису сюжетних ліній фільмів і почали їх опрацьовувати, використовуючи цю концепцію багатоміткової класифікації. І результати, навіть з використанням простої моделі, справді вражають.

У цьому розділі ми розглянемо практичний підхід до розуміння класифікації за кількома мітками в NLP.

Перед тим, як перейти до реалізації нашої програми в коді Phyton, нагадаємо концепцію багатоміткової класифікації в NLP. Важливо спочатку зрозуміти техніку, перш ніж заглиблюватися в реалізацію.

Основна концепція очевидна з назви – класифікація з кількома мітками. Тут екземпляр/запис може мати кілька міток, і кількість міток на екземпляр не є фіксованою.

Пояснимо це на простому прикладі. Погляньте на таблиці нижче (рис. 3.2), де «X» позначає вхідні змінні, а «y» – цільові змінні (які ми прогнозуємо).

Table 1

X	y
X_1	t_1
X_2	t_2
X_3	t_1
X_4	t_2
X_5	t_1

Binary Classification

Table 2

X	y
X_1	t_2
X_2	t_3
X_3	t_4
X_4	t_1
X_5	t_3

Multi-class Classification

Table 3

X	y
X_1	$[t_2, t_5]$
X_2	$[t_1, t_2, t_3, t_4]$
X_3	$[t_3]$
X_3	$[t_2, t_4]$
X_3	$[t_1, t_3, t_4]$

Multi-label Classification

Рисунок 3.2 – Принципи різних типів класифікації

«y» – це бінарна цільова змінна в Таблиці 1. Отже, є лише дві мітки – t1 та t2.

«y» містить більше двох міток у Таблиці 2. Але зверніть увагу, що для кожного вхідного значення в обох цих таблицях є лише одна мітка.

Таблиця 3 виділяється. Тут у нас є кілька тегів, не лише для всієї таблиці, але й для окремих вхідних даних.

Ми не можемо застосовувати традиційні алгоритми класифікації безпосередньо до такого набору даних. Чому? Тому що ці алгоритми очікують одну мітку для кожного вхідного значення, тоді як натомість у нас є кілька міток.

3.2 Налаштування задачі класифікації з кількома мітками

Існує кілька способів побудови системи рекомендацій. Коли йдеться про жанри фільмів, ви можете аналізувати дані на основі кількох змінних. Але ось простий підхід – побудуйте модель, яка може автоматично передбачати теги жанрів.

Наше завдання – побудувати модель, яка може передбачити жанр фільму, використовуючи лише деталі сюжету (доступні у текстовому вигляді).

Погляньте на знімок нижче з IMDb та виберіть різні елементи, що відображаються (див. рис. 3.3).

Рисунок 3.3 – Знімок екрану з інформацією про фільм в каталозі IMDb

У такому крихітному просторі багато інформації:

- Назва фільму.
- Рейтинг фільму у верхньому правому куті.
- Загальна тривалість фільму.
- Дата випуску.
- І, звичайно ж, жанри фільмів.

Жанри підказують нам, чого очікувати від фільму. А оскільки ці жанри клікабельні (принаймні на IMDb), вони дозволяють нам знаходити інші схожі

фільми того ж типу. Те, що здавалося простою функцією продукту, раптом має стільки багатообіцяючих опцій.

3.3 Про набір даних

Для нашого проєкту ми використовуватимемо відкритий набір даних CMU Movie Summary Corpus. Можна завантажити набір даних зі сторінки [11].

Цей набір даних містить кілька файлів, але поки що ми зосередимося лише на двох із них:

movie.metadata.tsv : Метадані для 81741 фільму, витягнуті з дампу Freebase. Теги жанру фільму доступні в цьому файлі.

plot_summaries.txt: Короткі описи сюжетів 42306 фільмів, взяті з дампу англомовної Вікіпедії. Кожен рядок містить ідентифікатор фільму Вікіпедії (який індексується в movie.metadata.tsv), а потім короткий опис сюжету.

3.4 Стратегія побудови моделі прогнозування жанрів фільмів

Ми знаємо, що не можемо використовувати алгоритми контрольованої класифікації безпосередньо на наборі даних з кількома мітками. Тому спочатку нам доведеться перетворити нашу цільову змінну. Давайте подивимося, як це зробити, використовуючи фіктивний набір даних (рис. 3.4).

X	y
X_1	$[t_2, t_5]$
X_2	$[t_1, t_3, t_2]$
X_3	$[t_4]$

Рисунок 3.4 – Набір даних для створення моделі

Тут X та y – це ознаки та мітки відповідно – це набір даних з кількома мітками. Тепер ми використаємо підхід бінарної релевантності для перетворення

нашої цільової змінної *y*. Спочатку ми видалимо унікальні мітки з нашого набору даних:

$$\text{Унікальні мітки} = [t1, t2, t3, t4, t5\ ]. \quad (3.1)$$

У даних є 5 унікальних тегів. Далі нам потрібно замінити поточну цільову змінну кількома цільовими змінними, кожна з яких належить до унікальних міток набору даних. Оскільки є 5 унікальних міток, буде 5 нових цільових змінних зі значеннями 0 та 1, як показано нижче (рис. 3.5).

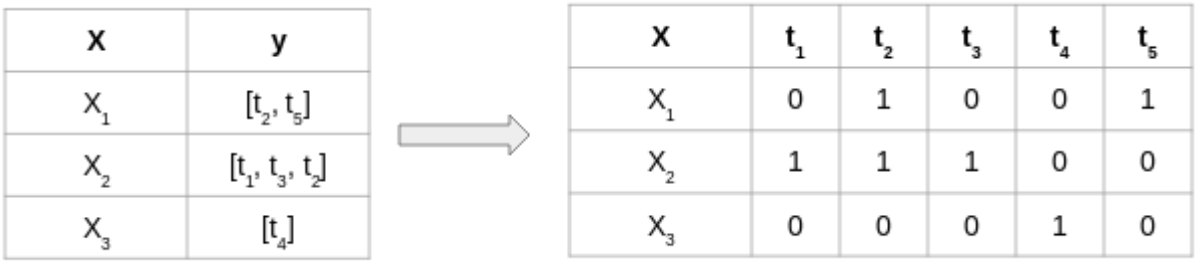

Рисунок 3.5 – Перетворення набору даних

Ми, таким чином, розглянули необхідну основу для вирішення цієї задачі.

3.5 Програмна реалізація

Реалізація – використання багатоміткової класифікації для побудови моделі прогнозування жанру фільму (на Python)

Ми зрозуміли постановку проблеми та розробили логічну стратегію для проектування нашої моделі. Давайте об'єднаємо все це разом.

Почнемо з імпорту бібліотек, необхідних для нашого проєкту.

Лістинг 3.1 – Імпорт бібліотек

```
import pandas as pd
import numpy as np
import json
import nltk
import re
import csv
import matplotlib.pyplot as plt
import seaborn as sns
from tqdm import tqdm
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split

%matplotlib inline
pd.set_option('display.max_colwidth', 300)
```

Спочатку завантажимо файл метаданих фільму. Використовуйте '\t' як роздільник, оскільки це файл, розділений табуляцією (.tsv) (лістинг 3.2):

Лістинг 3.2 – Завантаження набору даних

```
import pandas as pd
meta = pd.read_csv("movie.metadata.tsv", sep = '\t', header =
None)
print(meta.head())
```

Отриманий заголовок набору даних показано на рисунку 3.6.

	0	1	2	3	4	5	6	7	8
0	975900	/m/03vyhn	Ghosts of Mars	2001-08-24	14010832.0	98.0	{"/m/02h40lc": "English Language"}	{"/m/09c7w0": "United States of America"}	{"/m/01jfsb": "Thriller", "/m/06n90": "Science Fiction", "/m/03npn": "Horror", "/m/03k9fj": "Adventure", "/m/0fdjb": "Supernatural", "/m/02kdv5l": "Action", "/m/09zvmj": "Space western"}
1	3196793	/m/08yl5d	Getting Away with Murder: The JonBenét Ramsey Mystery	2000-02-16	NaN	95.0	{"/m/02h40lc": "English Language"}	{"/m/09c7w0": "United States of America"}	{"/m/02n4kr": "Mystery", "/m/03bxz7": "Biographical film", "/m/07s9rl0": "Drama", "/m/0hj3n01": "Crime Drama"}
2	28463795	/m/0crgdbh	Brun bitter	1988	NaN	83.0	{"/m/05f_3": "Norwegian Language"}	{"/m/05b4w": "Norway"}	{"/m/0lsxr": "Crime Fiction", "/m/07s9rl0": "Drama"}
3	9363483	/m/0285_cd	White Of The Eye	1987	NaN	110.0	{"/m/02h40lc": "English Language"}	{"/m/07ssc": "United Kingdom"}	{"/m/01jfsb": "Thriller", "/m/0glj9q": "Erotic thriller", "/m/09blyk": "Psychological thriller"}
4	261236	/m/01mrr1	A Woman in Flames	1983	NaN	106.0	{"/m/04306rv": "German Language"}	{"/m/0345h": "Germany"}	{"/m/07s9rl0": "Drama"}

Рисунок 3.6 – Набір даних

Тепер ми завантажимо набір даних сюжету фільму в пам'ять. Ці дані знаходяться в текстовому файлі, кожен рядок якого містить ідентифікатор фільму та сюжет фільму. Ми будемо зчитувати їх рядок за рядком (лістинг 3.3).

Лістинг 3.3 – Завантаження набору даних в пам'ять

```
plots = []

with open("plot_summaries.txt", 'r') as f:
        reader = csv.reader(f, dialect='excel-tab')
        for row in tqdm(reader):
             plots.append(row)
```

Далі розділимо ідентифікатори фільмів та сюжети на два окремі списки. Ми використаємо ці списки для формування фрейму даних (лістинг 3.4).

Лістинг 3.4 – Формування фрейму даних

```
movie_id = []
plot = []

# extract movie Ids and plot summaries
for i in tqdm(plots):
  movie_id.append(i[0])
  plot.append(i[1])

# create dataframe
movies = pd.DataFrame({'movie_id': movie_id, 'plot': plot})
```

Таким чином зміст датафрейму з відомостями про фільми може виглядати наступним чином (див. рис. 3.7).

	movie_id	plot
0	23890098	Shlykov, a hard-working taxi driver and Lyosha, a saxophonist, develop a bizarre love-hate relationship, and despite their prejudices, realize they aren't so different after all.
1	31186339	The nation of Panem consists of a wealthy Capitol and twelve poorer districts. As punishment for a past rebellion, each district must provide a boy and girl between the ages of 12 and 18 selected by lottery for the annual Hunger Games. The tributes must fight to the death in an arena; the sole...
2	20663735	Poovalli Induchoodan is sentenced for six years prison life for murdering his classmate. Induchoodan, the only son of Justice Maranchery Karunakara Menon was framed in the case by Manapally Madhavan Nambiar and his crony DYSP Sankaranarayanan to take revenge on idealist judge Menon who had e...
3	2231378	The Lemon Drop Kid , a New York City swindler, is illegally touting horses at a Florida racetrack. After several successful hustles, the Kid comes across a beautiful, but gullible, woman intending to bet a lot of money. The Kid convinces her to switch her bet, employing a prefabricated con. Unfo...
4	595909	Seventh-day Adventist Church pastor Michael Chamberlain, his wife Lindy, their two sons, and their nine-week-old daughter Azaria are on a camping holiday in the Outback. With the baby sleeping in their tent, the family is enjoying a barbecue with their fellow campers when a cry is heard. Lindy r...

Рисунок 3.7 – Оброблений датафрейм

Додамо назви фільмів та їхні жанри з файлу метаданих фільму, об'єднавши останній з першим на основі стовпця movie_id (див. рис. 3.8).

```
# change datatype of 'movie_id'
meta['movie_id'] = meta['movie_id'].astype(str)

# merge meta with movies
movies = pd.merge(movies, meta[['movie_id', 'movie_name', 'genre']], on = 'movie_id')

movies.head()
```

	movie_id	plot	movie_name	genre
0	23890098	Shlykov, a hard-working taxi driver and Lyosha, a saxophonist, develop a bizarre love-hate relationship, and despite their prejudices, realize they aren't so different after all.	Taxi Blues	{"/m/07s9rl0": "Drama", "/m/03q4nz": "World cinema"}
1	31186339	The nation of Panem consists of a wealthy Capitol and twelve poorer districts. As punishment for a past rebellion, each district must provide a boy and girl between the ages of 12 and 18 selected by lottery for the annual Hunger Games. The tributes must fight to the death in an arena; the sole...	The Hunger Games	{"/m/03btsm8": "Action/Adventure", "/m/06n90": "Science Fiction", "/m/02kdv5l": "Action", "/m/07s9rl0": "Drama"}
2	20663735	Poovalli Induchoodan is sentenced for six years prison life for murdering his classmate. Induchoodan, the only son of Justice Maranchery Karunakara Menon was framed in the case by Manapally Madhavan Nambiar and his crony DYSP Sankaranarayanan to take revenge on idealist judge Menon who had e...	Narasimham	{"/m/04t36": "Musical", "/m/02kdv5l": "Action", "/m/07s9rl0": "Drama", "/m/01chg": "Bollywood"}
3	2231378	The Lemon Drop Kid , a New York City swindler, is illegally touting horses at a Florida racetrack. After several successful hustles, the Kid comes across a beautiful, but gullible, woman intending to bet a lot of money. The Kid convinces her to switch her bet, employing a prefabricated con. Unfo...	The Lemon Drop Kid	{"/m/06qm3": "Screwball comedy", "/m/01z4y": "Comedy"}
4	595909	Seventh-day Adventist Church pastor Michael Chamberlain, his wife Lindy, their two sons, and their nine-week-old daughter Azaria are on a camping holiday in the Outback. With the baby sleeping in their tent, the family is enjoying a barbecue with their fellow campers when a cry is heard. Lindy r...	A Cry in the Dark	{"/m/0lsxr": "Crime Fiction", "/m/07s9rl0": "Drama", "/m/01f9r0": "Docudrama", "/m/03q4nz": "World cinema", "/m/05bh16v": "Courtroom Drama"}

Рисунок 3.8 – Опрацьований датафрейм

Ми не можемо отримати доступ до жанрів у цьому рядку, використовуючи лише .values() . Це тому, що цей текст є рядком, а не словником. Нам доведеться перетворити цей рядок у словник. Тут ми скористаємося допомогою бібліотеки json:

```
type(json.loads(movies['genre'][0]))
```

Тепер ми можемо легко отримати доступ до жанрів цього рядка:

```
json.loads(movies['genre'][0]).values()
```

Цей код допомагає нам витягти всі жанри з даних фільмів. Після цього додайте витягнуті жанри у вигляді списків назад до датафрейму даних фільмів (лістинг 3.5).

Лістинг 3.5 – Створення списку жанрів для кожного фільму

```
# an empty list
genres = []

# extract genres
for i in movies['genre']:
  genres.append(list(json.loads(i).values()))

# add to 'movies' dataframe
movies['genre_new'] = genres
```

Деякі зразки можуть не містити жодних тегів жанру. Нам слід видалити ці зразки, оскільки вони не відіграватимуть жодної ролі в процесі побудови нашої моделі. На рисунку 3.9 показано програмний код та результат його виконання.

```
# remove samples with 0 genre tags
movies_new = movies[~(movies['genre_new'].str.len() == 0)]

movies_new.shape, movies.shape
```

Output:

```
((41793, 5), (42204, 5))
```

Рисунок 3.9 – Програмний код та результат його виконання

Лише 411 фільмів не мають міток про жанри. Тому проаналізуємо наш датафрнейм ще раз (див. рис. 3.10).

	movie_id	plot	movie_name	genre	genre_new
0	23890098	Shlykov, a hard-working taxi driver and Lyosha, a saxophonist, develop a bizarre love-hate relationship, and despite their prejudices, realize they aren't so different after all.	Taxi Blues	{"/m/07s9rl0": "Drama", "/m/03q4nz": "World cinema"}	[Drama, World cinema]
1	31186339	The nation of Panem consists of a wealthy Capitol and twelve poorer districts. As punishment for a past rebellion, each district must provide a boy and girl between the ages of 12 and 18 selected by lottery for the annual Hunger Games. The tributes must fight to the death in an arena; the sole...	The Hunger Games	{"/m/03btsm8": "Action/Adventure", "/m/06n90": "Science Fiction", "/m/02kdv5l": "Action", "/m/07s9rl0": "Drama"}	[Action/Adventure, Science Fiction, Action, Drama]
2	20663735	Poovalli Induchoodan is sentenced for six years prison life for murdering his classmate. Induchoodan, the only son of Justice Maranchery Karunakara Menon was framed in the case by Manapally Madhavan Nambiar and his crony DYSP Sankaranarayanan to take revenge on idealist judge Menon who had e...	Narasimham	{"/m/04t36": "Musical", "/m/02kdv5l": "Action", "/m/07s9rl0": "Drama", "/m/01chg": "Bollywood"}	[Musical, Action, Drama, Bollywood]
3	2231378	The Lemon Drop Kid , a New York City swindler, is illegally touting horses at a Florida racetrack. After several successful hustles, the Kid comes across a beautiful, but gullible, woman intending to bet a lot of money. The Kid convinces her to switch her bet, employing a prefabricated con. Unfo...	The Lemon Drop Kid	{"/m/06qm3": "Screwball comedy", "/m/01z4y": "Comedy"}	[Screwball comedy, Comedy]
4	595909	Seventh-day Adventist Church pastor Michael Chamberlain, his wife Lindy, their two sons, and their nine-week-old daughter Azaria are on a camping holiday in the Outback. With the baby sleeping in their tent, the family is enjoying a	A Cry in the Dark	{"/m/0lsxr": "Crime Fiction", "/m/07s9rl0": "Drama", "/m/01f9r0": "Docudrama", "/m/03q4nz": "World cinema", "/m/05bb16v": "Courtroom	[Crime Fiction, Drama, Docudrama, World cinema, Courtroom

Рисунок 3.10 – Опрацьовані дані про фільми

Варто звернути увагу, що жанри тепер представлені у форматі списку. Наведений нижче код дозволяє встановити, скільки кіножанрів охоплено цим набором даних (рисунок 3.6).

```
# get all genre tags in a list
all_genres = sum(genres,[])
len(set(all_genres))
```

Output:

```
363
```

Рисунок 3.11 – Кількість жанрів по фільмах

У нашому наборі даних понад 363 унікальних тегів. Це досить велика кількість. Давайте дізнаємося, що це за теги. Ми використаємо FreqDist () з бібліотеки nltk для створення словника жанрів та кількості їх появи в наборі даних (лістинг 3.7).

Після цього виконаємо операції з очищення тексту, які не мають прямого стосунку до нашої роботи.

Лістинг 3.6 – Очищення текстової інформації про фільми

```
# function for text cleaning
def clean_text(text):
    # remove backslash-apostrophe
    text = re.sub("\'", "", text)
    # remove everything except alphabets
    text = re.sub("[^a-zA-Z]"," ",text)
    # remove whitespaces
    text = ' '.join(text.split())
    # convert text to lowercase
    text = text.lower()

    return text
```

Лістинг 3.7 – Підрахунок унікальних значень тегів для жанру фільму

```
all_genres = nltk.FreqDist(all_genres)

# create dataframe
all_genres_df = pd.DataFrame({'Genre': list(all_genres.keys()),
                              'Count': list(all_genres.values())})
```

Відобразимо розподіл жанрів графічно (див. рис. 3.12). Програмний код показаний у лістингу нижче:

Лістинг 3.7 – Побудова діаграми розподілу жанрів

```
g = all_genres_df.nlargest(columns="Count", n = 50)
plt.figure(figsize=(12,15))
ax = sns.barplot(data=g, x= "Count", y = "Genre")
ax.set(ylabel = 'Count')
plt.show()
```

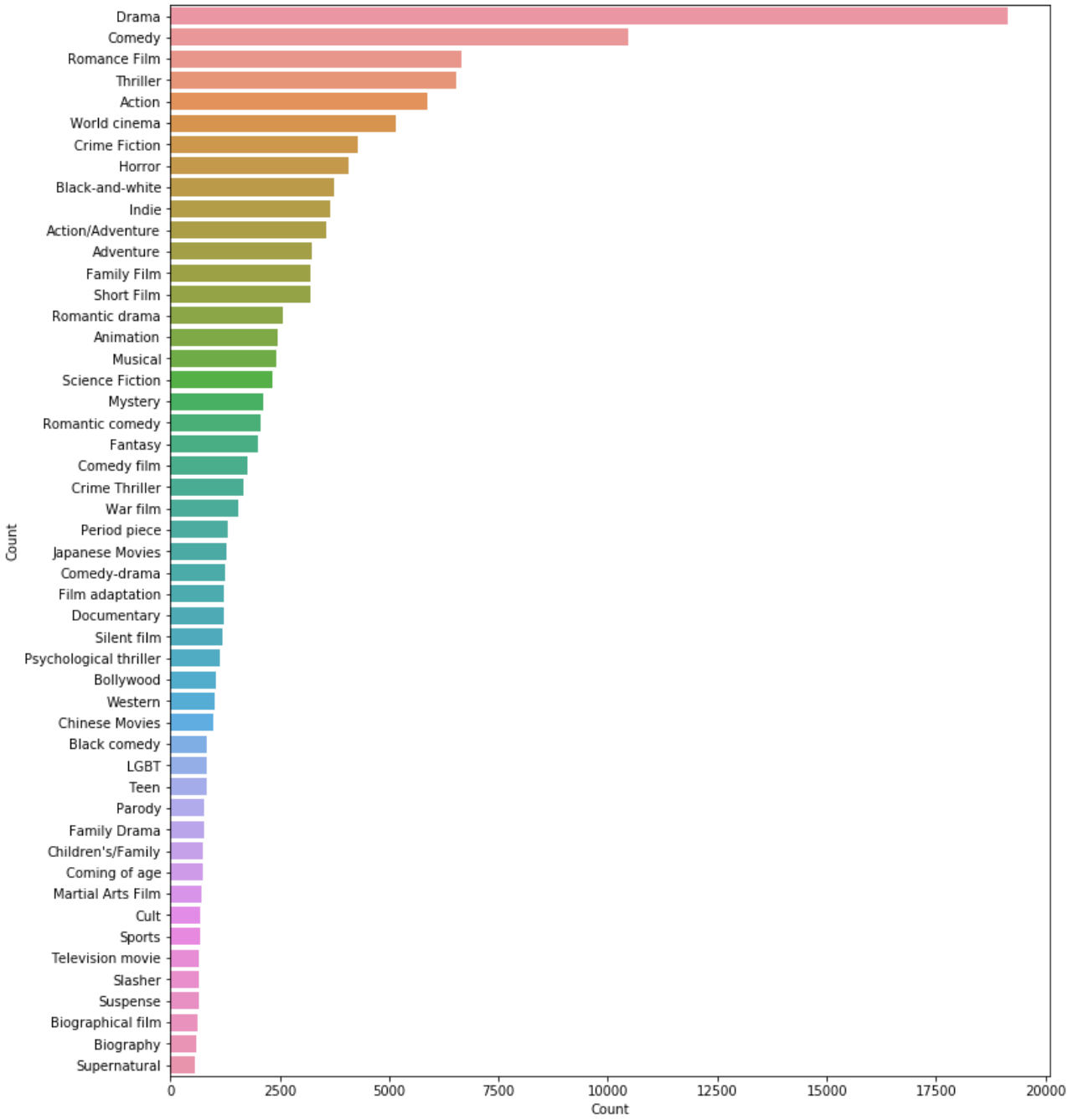

Рисунок 3.12 – Частота згадувань жанрів фільмів

3.6 Перетворення тексту на функції

Як згадувалось раніше, що ми розглядатимемо цю задачу класифікації з кількома мітками як проблему бінарної релевантності. Отже, тепер ми швидко закодуємо цільову змінну, тобто genre_new, використовуючи sklearn MultiLabelBinarizer (). Оскільки існує 363 унікальних тегів жанру, з'явиться 363 нових цільових змінних.

Лістинг 3.8 – Створення цільових змінних

```
from sklearn.preprocessing import MultiLabelBinarizer

multilabel_binarizer = MultiLabelBinarizer()
multilabel_binarizer.fit(movies_new['genre_new'])

# transform target variable
y = multilabel_binarizer.transform(movies_new['genre_new'])
```

Тепер настав час зосередитися на вилученні ознак з очищеної версії даних сюжетів фільмів. Будемо використовувати ознаки TF-IDF. Можна використовувати будь-який інший метод вилучення ознак, наприклад, Bag-of-Words, word2vec, GloVe або ELMo .

```
tfidf_vectorizer = TfidfVectorizer ( max_df = 0.8, max_features = 10000)
```

Ми використали 10 000 найчастіших слів у даних як свої ознаки. Також можна спробувати будь-яке інше число для параметра max_features .

Тепер, перед створенням функцій TF-IDF, ми розділимо наші дані на навчальні та валідаційні набори для навчання та оцінки продуктивності нашої моделі. Традиційно застосовується поділ 80-20, де 80% зразків даних у навчальному наборі, а решта у валідаційному:

```
# split dataset into training and validation set
xtrain, xval, ytrain, yval = train_test_split(movies_new['clean_plot'], y, test_size=0.2, random_state=9)
```

Тепер ми можемо створити функції для навчального набору даних та набору валідації:

```
# create TF-IDF features
xtrain_tfidf = tfidf_vectorizer.fit_transform(xtrain)
xval_tfidf = tfidf_vectorizer.transform(xval)
```

Ми готові до складання моделі.

Нам доведеться побудувати модель для кожної цільової змінної, закодованої за одним кодом. Оскільки у нас є 363 цільові змінні, нам доведеться підібрати 363 різні моделі з однаковим набором предикторів (ознак TF-IDF).

Як вможна уявити, навчання моделей 363 може зайняти значний час на скромній системі. Тому побудуємо модель логістичної регресії, оскільки вона швидко навчається на обмеженій обчислювальній потужності (лістинг 3.9):

Лістинг 3.9 – Побудова логістичної регресії

```
from sklearn.linear_model import LogisticRegression

# Binary Relevance
from sklearn.multiclass import OneVsRestClassifier

# Performance metric
from sklearn.metrics import f1_score
```

Ми будемо використовувати sk-learn Клас OneVsRestClassifier для вирішення цієї задачі як проблеми бінарної релевантності або задачі «один проти всіх»:

```
lr = LogisticRegression()
clf = OneVsRestClassifier(lr)
```

Нарешті, застосовуємо модель до тренувального набору даних:

```
# fit model on train data
clf.fit(xtrain_tfidf, ytrain)
```

Передбачимо жанри фільмів на валідаційному наборі:

```
# make predictions for validation set
y_pred = clf.predict(xval_tfidf)
```

Давайте розглянемо зразок цих прогнозів (див. рис. 3.13).

```
array([0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Рисунок 3.13 – Робота моделі з валідаційним набором даних

Це двійковий одновимірний масив довжиною 363. По суті, це одноразово закодована форма унікальних тегів жанру. Нам доведеться знайти спосіб перетворити її на теги жанру фільму.

Нам знову на допомогу приходить sk-learn. Ми використаємо функцію inverse_transform () разом з об'єктом MultiLabelBinarizer (), щоб перетворити передбачувані масиви на теги жанру фільму (рисунок 3.14).

```
multilabel_binarizer.inverse_transform(y_pred)[3]
```

Output:

```
('Action', 'Drama')
```

Рисунок 3.14 – Перетворення числових міток у текстові значення

Однак, щоб оцінити загальну продуктивність нашої моделі, нам потрібно врахувати всі прогнози та всю цільову змінну набору валідації (див. рис. 3.15).

```
# evaluate performance
f1_score(yval, y_pred, average="micro")
```

Output:

```
0.31539641943734015
```

Рисунок 3.15 – Оцінка продуктивності моделі

Ми отримуємо пристойний бал F1 – 0,315. Ці прогнози були зроблені на основі порогового значення 0,5, що означає, що ймовірності, більші або рівні 0,5, були перетворені на одиниці, а решта – на нулі.

Давайте спробуємо змінити це порогове значення та подивимося, чи покращить це оцінку нашої моделі (лістинг 3.10).

```
# predict probabilities
y_pred_prob = clf.predict_proba(xval_tfidf)
```

Тепер встановимо порогове значення:

```
t = 0.3 # threshold value
y_pred_new = (y_pred_prob >= t).astype(int)
```

Ми використали 0,3 як порогове значення. Також варто спробувати інші значення. Давайте ще раз перевіримо результат значення F1 за цими новими прогнозами (див. рис. 3.16).

```
# evaluate performance
f1_score(yval, y_pred_new, average="micro")
```

Output:

```
0.4378456703198025
```

Рисунок 3.16 – Результати для нового порогового значення

Це досить значне покращення продуктивності нашої моделі. Кращим підходом для пошуку правильного порогового значення було б використання k-кратної перехресної перевірки та спроба різних значень.

Останнім кроком створимо функцію виводу. Нам також потрібно враховувати нові дані або нові сюжети фільмів, які з'являться в майбутньому. Наша система прогнозування жанрів фільмів повинна мати змогу приймати сюжет фільму в необробленому вигляді як вхідні дані та генерувати теги його жанру.

Щоб досягти цього, створимо функцію виводу. Вона візьме сюжетний текст фільму та виконає такі кроки:

1. Очистити текст.
2. Видалити стоп-слова з очищеного тексту.
3. Виокремити ознаки з тексту.
4. Прогнозування.
5. Повернути передбачені теги жанру фільму.

Програмний код показаний в лістингу 3.10.

Лістинг 3.10 – Функція виводу

```
def infer_tags(q):
    q = clean_text(q)
    q = remove_stopwords(q)
    q_vec = tfidf_vectorizer.transform([q])
    q_pred = clf.predict(q_vec)
    return multilabel_binarizer.inverse_transform(q_pred)
```

Давайте перевіримо цю функцію виведення на кількох зразках з нашого набору даних.

```
for i in range(5):
  k = xval.sample(1).index[0]
  print("Movie: ", movies_new['movie_name'][k], "\nPredicted genre: ", infer_tags(
  xval[k])), print("Actual genre: ",movies_new['genre_new'][k], "\n")
```

Отримаємо результат, показаний на рисунку 3.17.

```
Movie:  The Other Me
Predicted genre:  [('Comedy', 'Family Film')]
Actual genre:  ['Family Film', 'Fantasy', 'Comedy']

Movie:  Paperman
Predicted genre:  [()]
Actual genre:  ['Short Film', 'Animation']

Movie:  Teheran 43
Predicted genre:  [('Drama',)]
Actual genre:  ['Thriller', 'Crime Fiction', 'Drama', 'War film', 'Romance Film', 'Action']

Movie:  To Aru Hikūshi e no Tsuioku
Predicted genre:  [()]
Actual genre:  ['Anime']

Movie:  Half Human
Predicted genre:  [('Horror',)]
Actual genre:  ['Japanese Movies', 'Science Fiction', 'Horror', 'Creature Film']
```

Рисунок 3.17 – Результати роботи функції виводу

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці та її актуальність в ІТ-сфері

Для підвищення ефективності системи управління охорони праці (СУОП) дуже важлива роль належить формуванню і розвитку інформаційної культури фахівців ІТ-технологій, яка впливає на удосконалення інформаційного контуру сучасних підприємств, дозволяє створювати надійні прогнози щодо стану умов праці, показників здоров'я та працездатності, виробничого травматизму і професійної захворюваності, визначати політику розвитку підприємств, установ та організацій на основі різноманітних стратегій охорони праці (інноваційні, маркетингові, інвестиційні, фінансові, технологічні, диверсифікаційні). Поряд з інформаційною культурою важливо використовувати в рамках СУОП «трикутник» її складових: правову, організаційну, управлінську.

В управлінні охороною праці потрібно реалізувати основні положення, окремі теоретико-методологічні підходи інформаційного менеджменту. Головну роль та відповідальність за стан СУОП мають нести фахівці служби охорони праці сучасного підприємства.

Сучасне суспільство називають постіндустріальним, постекономічним, інформаційним, оскільки йдеться про багатосторонні і кардинальні зміни у розвитку цивілізації.

Інформаційне суспільство передбачає докорінну зміну, яка полягає у перетворенні інформації і знань у головний професійно-виробничий потенціал особистості, соціуму і держави.

На постіндустріальному етапі розвитку суспільства вирішальним фактором стає інформація. Її домінування ініціювала науково-технічна революція, яку ще іменують інформаційною, оскільки нею охоплена будь-яка інтелектуальна діяльність, починаючи з інформаційних образів штучного

інтелекту у нових технологіях, економіки, і продовжуючи інформатизацією суспільства в умова світової глобалізації науки й освіти тощо.

Інформаційні технології розглядаються як потужний важіль економічного зростання України. Для цього необхідні значні стратегічні інвестиції у комп'ютерну та комунікаційну інфраструктуру, програми досліджень і розробок, освітню галузь [12].

Під інформаційною культурою розуміють сукупність, складову НІТ (новітні інформаційні технології), технологічну, правову, психологічну, соціологічну та ергономічну підсистеми, що сприяють спрямованому впливу на протікання соціальних процесів у суспільстві, колективі і вихованню свідомого відношення людини до праці, виконання прав та обов'язків [12].

Поняття інформаційної культури виникло в процесі активізації дослідницької уваги до механізмів інформаційного обміну у зв'язку зі значним підвищення ролі інформації в соціокультурних процесах суспільства, яке розглядають як інформаційне суспільство знань, де в центрі знаходяться інформаційні технології.

Робота з інформацією та інформаційна культура в цілому є одним з найважливіших компонентів спроб компанії управляти змінами. Є три принципові причини, в силу яких сьогодні необхідно дбати про інформаційну культуру компанії.

По-перше, вона все більше і більше стає найважливішою частиною загальної організаційної (корпоративної) культури компанії. Все більше компаній розуміють необхідність перетворень, орієнтованих на задоволення очікувань споживача. Щоб сьогодні впливати на майбутнє, потрібно уявляти собі на що вона буде схожа. А для цього потрібно працювати з різноманітною діловою, професійною, технологічною, соціальною, ринковою та політичною інформацією.

По-друге, інформаційні технології роблять можливим створення в компаніях комп'ютерних мереж, за допомогою яких йде спілкування між менеджерами, але важливо знати, як люди використовують цю інформацію. Саме

по собі створення такої мережі з усіма її робочими станціями і мультимедійними можливостями не гарантує того, що інформація буде використовуватися більш розумно і більш ефективно.

По-третє, для різних функціональних служб, підрозділів та робочих груп сучасних підприємств в сфері охорони праці інформаційна культура різна, а це означає відмінність методологічних підходів до процесів усвідомлення, збору, організації, обробки, поширення і використання інформації. Тому багато менеджерів погодяться з тим, що корпоративна інформаційна культура важлива для вироблення різних стратегій охорони праці та запровадження відповідних заходів з її вдосконалення.

Для деяких галузей, таких як розробка програмного забезпечення, інформаційна культура є необхідною умовою виживання, тому що зміна технологій в розробці програмного забезпечення відбувається кожні 6-8 місяців, а інвестиції на підготовку персоналу і освоєння нової технології величезні і у великих компаніях варіюються від 1,5 до 2 млрд. доларів на рік [13].

Аналіз свідчить, що інформатизація та інтеграція комунікаційного простору України сприяє різкому підвищенню інформаційної та професійної компетентності, ділової активності, стимулюванню конкуренції, створенню інноваційних підприємств та організацій, нових робочих місць, зниженню витрат на утримання управлінського апарату [13].

Поряд із задачами і здобутками окреслилися негативи використання інформаційних технологій:

1) надмірне інформаційне навантаження, суть якого полягає у тому, що кількість корисної інформації, яка надходить до мережі, перевищує психофізіологічні можливості її сприйняття людиною;

2) велика кількість інформації, яка сприймається, але не є корисною для фахівців в даний момент;

3) інформаційний голод, причиною якого є саме надлишок інформації, викликаний інформаційним перенавантаженням;

4) «інформоманія» як хвороба людини, яка робить останню знеособленою, залежною від перебування в інформаційному просторі і роботи з комп’ютером і чому вона віддає перевагу, уникаючи «живого» спілкування з людьми;

5) поява «кіберспільнот», що за своїми соціокультурними характеристиками набагато ближчі до представників інших культур у глобальному інформаційному просторі, ніж до своєї етнонаціональної спільноти чи решти населення, не охопленого Інтернетом;

6) індивідуалізм і дегуманізація способу життя «мешканців» Інтернету – відсутність готовності ділитися своїми знаннями.

Слід розуміти, що комп'ютерні технології, а особливо їх мережі істотно впливають на життєдіяльність людини, припускаючи глобалізацію і технократизацію суспільства. Але в ще більшій мірі цей вплив поширюється безпосередньо на центральну нервову систему, яка звикає працювати в дуже інтенсивному режимі багатозадачності, де вже переважають не тривалі логічні роздуми, а інтуїтивно-реактивні ланцюжки розумових формулювань у зв'язку з величезним обсягом оброблюваної щодня інформації, кількість якої зростає за експоненціальною швидкістю. Виникає припущення, що саме збільшення обсягу інформації та прискорення її обробки людиною може згубно вплинути на розвиток розумових здібностей людини.

Аналіз продуктивності розумової праці в найбільших за чисельністю фахівців IT-фірм показав, що велике значення з точки зору впливу на її результати має організаційна (корпоративна) культура. В цьому напрямі влаштовуються різні тімбілдинги, заходи, тренінги для розвитку персоналу. Також кожен керівник повинен добре розуміти свого співробітника, що саме для нього важливо, що його мотивує. Важливо відвести потрібну роль відповідному співробітнику, щоб він виконував ті завдання, які йому цікаві.

На подібних тренінгах в тому числі повинна розглядатися інформаційна культура працівника, в освоєнні, володінні, мотивуванні, застосуванні, перетворенні інформації із застосуванням сучасних інформаційних технологій і використанням цих умінь в навчанні з охорони праці і в подальшій професійній

діяльності. Особливо вони будуть корисні, як доповнення до існуючих інструктажів з охорони праці на підприємстві, або як контроль психологічного стану та взаємовідносин у колективі.

Інформаційна культура як інтегративне утворення абсолютно не зводиться до розрізнених знань, вмінь та навичок роботи за комп'ютером. Вона передбачає інформативну спрямованість цілісної особистості, яка володіє мотивацією до застосування і засвоєння нових даних. Інформаційну культуру можна розглядати, як одну з граней особистісного розвитку промислових робітників. Це шлях універсалізації якостей людини.

Оволодіння інформаційною культурою сприяє реальному розумінню особистістю свого місця, себе і своєї ролі у виробничому колективі. Вона має сприяти формуванню нового покоління фахівців інформаційного суспільства, який повинен володіти наступними навичками: виділення релевантної, значущої інформації, диференціації вихідних даних, розробки інформативних критеріїв її оцінки інформації, вміло використовувати її в рамках СУОП.

Сьогодні продовжує діяти стратегічне правило «Можливості комп’ютерної техніки обмежені тільки нашими уявленнями» [14, 15].

4.2 Шкідлива дія шуту та вібрації і захист від неї

Для запобігання шкідливої дії шуму і вібрації на організм працюючих проводяться технічні, організаційні і медикопрофілактичні заходи.

Одним з основних технічних заходів є зменшення при експлуатації та на стадії проектування, конструювання обладнання причин шуму і вібрації в самому джерелі утворення. Досягають цього завдяки використанню раціональної конструкції обладнання, заміни ударної дії деталей і машин коливальною, з’єднання елементів гнучкими зв’язками, врівноважування обертових частин механізмів, заміни металевих деталей пластмасовими, забезпечення різних власних частот коливань механізму з частотою збуджуючої сили. Аеродинамічний шум може бути зменшений застосуванням глушників та

повітропроводів зі змінним перерізом. Шум трансформаторів (електромагнітний шум) знижується, якщо застосувати листи заліза як складових осердя трансформатора з малою магнітострикцією, серцевини.

Якщо неможливо ізолювати чи знизити шум і вібрацію самого джерела, потрібно:

– ізолювати джерело шуму або вібрації від навколишнього середовища засобами вібро- та звукоізоляції

– раціонально планувати виробничі приміщення, що мають інтенсивні джерела шуму;

– збільшувати звукопоглинання внутрішніх поверхонь приміщення шляхом звукопоглинальних покриттів.

Принцип роботи звукоізоляційних екранів оснований на відбиванні звукової хвилі від різних екранів, стін, кожухів обладнання. Шумливі агрегати слід закривати звукоізоляційними кожухами з виводом назовні органів керування та контрольних приладів. Звукоізоляційні екрани виготовляють з металу, деревини, пластмаси та інших щільних матеріалів. Екрани зсередини покривають звукопоглинаючими матеріалами (скловатою пінополіуретаном), а по периметру кожуха – віброізоляційними підкладками (гума).

Вихідними даними для розрахунку параметрів необхідного екрану є спектр шуму, який необхідно ослабити, кількість екранів, через які проходить шум, їх площа, акустичні характеристики приміщення.

За розрахованими значеннями необхідної звукової ізоляційної здатності екрану підбирається матеріал конструкції й екрану.

Принцип звукопоглинання оснований на явищі трансформації коливальної енергії звуку в теплову через втрати при терті. Найбільші втрати при терті мають пористі, волокнисті і перфоровані матеріали: поролон, пемзолітові і деревоволокнисті плити тощо.

Енергія звукової хвилі переходить у теплову енергію, причому, ефект звукоізоляції збільшується з ростом частоти звукової хвилі. Звукопоглинаючими матеріалами оббивають стелі, стіни. Щоб одержати ефективну звукоізоляцію,

найбільш доцільно застосовувати багатошарові огородження з м’якими прошарками (мінеральна вата).

Важливим технічним рішенням у забезпеченні виробничих умов є вдосконалення ручних віброінструментів. Для цього використовують віброгасіння, змінюють ударний вузол, проводять балансування частин, що обертаються.

Послаблення локальної вібрації і передачі вібрації на підлогу і сидіння досягається засобами віброізоляції і вібропоглинання, застосуванням пружинних і гумових амортизаторів, прокладок тощо. Для обмеження поширення вібрацій через ґрунт, між фундаментом і ґрунтом залишають повітряні проміжки, які називаються акустичними розривами.

В останні роки знаходять застосування динамічні віброгасники, в яких створюються вібрації, що співпадають по частоті і протилежні по фазі вібрації машини, коливання якої необхідно зменшити.

До організаційних заходів по боротьбі з шумом та вібрацією на виробництві відносяться: впровадження раціонального режиму праці і відпочинку, обмеження часу роботи при використанні ручного інструменту, який створює вібрацію.

Глушники звуку застосовуються для зменшення шуму аеродинамічних установок (вентиляторів, пневмоінструментів, газотурбінних, дизельних, компресорних установок). Вони поділяються на активні, які поглинають звукову енергію, що на них поступила, і реактивні, які відбивають цю енергію. Потужні джерела шуму як правило розміщують в окремих приміщеннях, які віддалені від постійних робочих місць.

Ізоляційні кабіни або екрани застосовують як екрани робочих місць для зменшення зовнішніх шумів.

Якщо не вдається зменшити рівень шуму і вібрації на робочому місці до нормативних значень та необхідно використовувати засоби індивідуального захисту: рукавиці, взуття, навушники, м’які шоломи, які зменшують рівень звукового тиску на 40-50 дБ.

У процесі виробництв, експлуатації і зберігання радіоелектронних засобів можуть виникати механічні і динамічні дії, що характеризуються широким діапазоном частот коливань, а також амплітудою, прискоренням і часом дії. Рівень механічних дій визначається умовами транспортування й експлуатації.

Необхідно розрізняти два види механічних дій: удари і вібрації. Удар виникає, коли апаратура отримує швидку зміну прискорення (піддаються удару входи кабелів, джгути, резистори, конденсатори, напівпровідникові діоди і тріоди, силові трансформатори, дроселі тощо). Вібрації – довготривалі знакозмінні процеси, які впливають на роботу апаратури при безпосередньому контакті з джерелом коливань або через повітряне середовище.

У результаті дії вібрацій і удару можуть бути наступні ушкодження апаратури: порушення герметичності через псування паяльних, зварних і клеєних швів і появи тріщин у метало-скляних спаях; повне руйнування корпусів або окремих їх частин через механічний резонанс або циклічну втому; обривання монтажних зв’язків, відшарування багатошарових друкованих плат, руйнування підставок; вихід з ладу електричних контактів; модуляція розмірів хвилеводних трактів; коаксіальних кабелів, конденсаторів змінної ємності, коливальних контурів, електровакуумних приладів, зміщення положення органів настроювання і управління.

Під впливом вібрацій може статись зміна параметрів напівпровідникових приладів, вольт амперних характеристик діодів, транзисторів. Все це призводить до руйнування конструкцій за рахунок явищ втоми.

Радіоелектронна апаратура (РЕА) повинна мати віброміцність, вібростійкість, ударостійкість.

Захист РЕА здійснюється наступними групами методів:

– зменшується інтенсивність джерел вібрації шляхом балансування, зменшення зазорів, віброізоляції джерела вібрацій;

– зменшується величина дій, що передається апаратом шляхом віброізоляції, демпфірування, виключення резонансів, активного віброзахисту за допомогою ексцентриків, маятників, гіроскопів;

– використання найбільш добротні і жорсткі компоненти і вузли;

– застосовуються амортизатори.

Захист часом, захист віддалю, усунення джерела тепловиділення, теплоізоляція, охолодження гарячої поверхні, забезпечення тепловіддачі тіла людини та індивідуальні засоби захисту.

Захист часом передбачає обмеження часу перебування робітника в зоні дії інфрачервоного випромінювання. Потужність випромінювання можна знизити за рахунок конструкторських і технологічних рішень (зміною нагрівання виробів у нагрівальних пічках індукційним нагріванням та ін.) і за рахунок покриття поверхні, яка нагрівається, тепло ізолювальним матеріалом.

Якщо теплоізоляція неможлива, тоді захист від прямої дії інфрачервоного випромінювання здійснюється екрануванням.

Екрани можуть бути прозорими, напівпрозорими і непрозорими.

У свою чергу вони поділяються на тепловідбивальні, тепловідвідні та теплопоглинальні; стаціонарні і нестаціонарні.

Застосовують також прозору водяну завісу у вигляді суцільної тонкої водяної плівки. Вода є активним поглиначем інфрачервоного випромінювання.

Перегрівання людини попереджують раціональним режимом пиття, режимом праці та гідро процедурами. Спецодяг виготовляється з незаймистого, стійкого до інфрачервоного випромінювання, м’якого і повітронепроникного матеріалу (тканина з металевим покриттям відбиває 90% інфрачервоного випромінювання).

Для захисту очей застосовують світлофільтри зі спеціального жовто-зеленого або синього скла.

Першочергові заходи – це конструкторські і технологічні рішення, які виключають генерацію або понижують інтенсивність випромінювання. Спеціальні засоби захисту (екранування джерел випромінювання, фарбування стін у світлі кольори) попереджують розповсюдження і знижують інтенсивність цих випромінювань у виробничих приміщеннях. Очі захищають окулярами або щитками зі склом – світлофільтром. Для захисту шкіри використовують мазі з

речовинами – світлофільтрами для цих променів (салол, саліцилово-метиловий ефір та ін.), а також спецодяг з бавовняних тканин і грубововняного сукна. Руки захищають рукавицями.

ВИСНОВКИ

У цій кваліфікаційній роботі було здійснено всебічне дослідження проблематики багатофакторної (багатоміткової) класифікації фільмів із використанням сучасних методів машинного навчання. Основною метою було розробити систему, здатну ефективно визначати жанрову належність фільмів, зважаючи на те, що один фільм може одночасно відноситися до кількох жанрів. Це завдання має практичну цінність, зокрема в контексті рекомендаційних систем, кіноархівів та пошукових сервісів.

Перший розділ було присвячено теоретичному аналізу підходів до багатоміткової класифікації. Розглянуто різницю між бінарною, багатокласовою та багатофакторною класифікацією. Особливу увагу приділено проблемам незбалансованості класів, кореляції між мітками та розмитості меж між жанрами. Було виявлено, що найбільш поширеними підходами є методи трансформації задачі, зокрема бінарна релевантність, та методи адаптації алгоритмів, які дозволяють працювати з кількома мітками безпосередньо.

У другому розділі здійснено огляд предметної області. Проаналізовано практики використання багатофакторної класифікації у реальних задачах, таких як категоризація тексту, медична діагностика, аналіз музики та сцен. Показано, що багатоміткова класифікація має широку сферу застосування, і її використання стрімко зростає. У контексті фільмів багатомітковість відображає природну багатожанровість кінострічок, що потребує гнучких та точних моделей класифікації.

У третьому розділі було розроблено та реалізовано програмну систему класифікації фільмів за жанрами на основі сюжетного опису. Використано відкритий набір даних CMU Movie Summary Corpus, проведено попередню обробку текстів, зокрема очищення та токенізацію, після чого виконано перетворення жанрових тегів у формат багатоміткової бінаризації. Для вилучення ознак застосовано метод TF-IDF, а для моделювання – логістичну

регресію з підходом One-vs-Rest. Було отримано F1-оцінку 0,315, що вказує на прийнятну точність навіть при використанні простої моделі.

Отже, виконана робота підтвердила ефективність підходів машинного навчання у вирішенні задачі багатоміткової класифікації фільмів. Результати показали, що навіть базові алгоритми здатні забезпечити прийнятний рівень точності, а подальше удосконалення моделі (зокрема, через використання глибокого навчання) може значно підвищити її продуктивність.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сороківська, О. Constructing a model of dual professional competences in engineering education [Електронний ресурс] / Олена Сороківська, Ірина Струтинська, Мельник Лілія, Ольга Мишкович // Соціально-економічні проблеми і держава. — 2021. — Вип. 2 (25). — С. 433-447. — Режим доступу: http://sepd.tntu.edu.ua/images/stories/pdf/2021/21soaiee.pdf.

2. Yukalo, V., Storozh, L., Datsyshyn, K., & Krupa, O. (2018). ELECTROPHORETIC SYSTEMS FOR PREPARATIVE FRACTIONATION OF PROTEIN PRECURSORS OF BIOACTIVE PEPTIDES FROM COW'S MILK. Food Science and Technology, 12(2). https://doi.org/10.15673/fst.v12i2.932.

3. Tsoumakas, G., & Katakis, I. (2008). Multi-label classification: An overview. Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications, 64-74.

4. Boutell, M.R., Luo, J., Shen, X. & Brown, C.M. (2004), 'Learning multi-label scene classification', Pattern Recognition, vol. 37, no. 9, pp. 1757-71.

5. knowledgator/events_classification_biotech · Datasets at Hugging Face. Hugging Face – The AI community building the future. URL: https://huggingface.co/datasets/knowledgator/events_classification_biotech (date of access: 21.05.2025).

6. Clare, A. & King, R.D. (2001), 'Knowledge Discovery in Multi-Label Phenotype Data', paper presented to Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2001), Freiburg, Germany.

7. Diplaris, S., Tsoumakas, G., Mitkas, P. & Vlahavas, I. (2005), 'Protein Classification with Multiple Algorithms', paper presented to Proceedings of the 10th Panhellenic Conference on Informatics (PCI 2005), Volos, Greece, November.

8. Jin, R. & Ghahramani, Z. (2002), 'Learning with Multiple Labels', paper presented to Proceedings of Neural Information Processing Systems 2002 (NIPS 2002), Vancoucer, Canada.

9. Li, T. & Ogihara, M. (2003), 'Detecting emotion in music', paper presented to Proceedings of the International Symposium on Music Information Retrieval, Washington D.C., USA.

10. Schapire, R. E., & Singer, Y. (2000). BoosTexter: A boosting-based system for text categorization. Machine learning, 39, 135-168.

11. CMU Movie Summary Corpus. CMU School of Computer Science. URL: https://www.cs.cmu.edu/~ark/personas/ (date of access: 21.04.2025).

12. Стручок, В. С., Стручок, О. С., & Мудра, Д. В. (2017). Навчальний посібник до написання розділу дипломного проекту та дипломної роботи ″Безпека в надзвичайних ситуаціях ″для студентів всіх спец. денної, заочної (дистанційної) та екстернатної форм навчання.

13. Стручок, В. С. (2022). Техноекологія та цивільна безпека. Частина "Цивільна безпека". Навчальний посібник.

14. Жидецький, В. Ц., Джигирей, В. С., & Мельников, О. В. (2000). Основи охорони праці. Львів: Афіша, 350, 132-136.

15. Навакатікян, О. О., Кальниш, В. В., & Стрюков, С. М. (1997). Охорона праці користувачів комп’ютерних відеодисплейних терміналів. О. Навакатікян.