УДК 004.415.2
DOI https://doi.org/10.32782/2663-5941/2024.2/16

Олещенко Л.М.
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ДАНИХ ПЛАТФОРМИ YOUTUBE

У цій статті розглянуто програмні інструменти, призначені для аналізу даних платформи YouTube, наведено огляд їх функціональності та основних можливостей. Розроблено програмне забезпечення, яке дозволяє виконувати статистичний аналіз даних про відео, виділяти тематики відео здійснювати інтелектуальний аналіз даних, який може бути корисним для аналітиків, маркетологів та створювачам контенту.

Графічний інтерфейс програмного забезпечення розроблено за допомогою інструментів мови програмування Python, використано середовище Jupyter-Notebook, бібліотеки NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, TensorFlow. Реалізована програмна система дозволяє опрацьовувати дані у вигляді таблиць csv, в першу чергу, з сервісу Kaggle, звідки було узято дані для виконання аналізу даних.

У результаті дослідження створено програмне забезпечення для роботи з даними, для операцій відкриття, зчитування, інтерпретації та візуалізації даних. Програмне забезпечення складається з окремих функціональних блоків, кожен з яких містить невелику кількість програмного коду, та відповідає за виконання певного обмеженого набору функцій з завантаження та підготовки даних, а також їх подальшого аналізу різними методами. Реалізовані методи для інтелектуального аналізу даних, такі як лінійна регресія, розрахунок кореляцій, агрегація табличних даних.

Набір даних складається з понад 350 000 годин відео. Для зменшення місць зберігання та обчислень використовано заздалегідь обчислені та стислі функції, які дають змогу тренувати модель на наборі даних менше, ніж за день на одному графічному процесорі. Відеоролики попередньо були оброблені для отримання 1.3 мільярдів візуальних функцій та 1.3 мільярдів аудіофункцій. Було витягнуті функції на рівні відео, а також функції на рівні кадрів та сегментів (при роздільній здатності 1 секунди). Візуальні ознаки були вилучені за допомогою моделі анотацій зображення Inception-V3, підготовленої на ImageNet. Звукові функції були витягнуті за допомогою звукової моделі VGG на попередній версії YouTube-8M. І візуальні, і аудіофункції були оброблені алгоритмом PCA та квантовані для розміщення на одному жорсткому диску. Комбінований набір усіх функцій має розмір менше 2 ТБ. Лексика цільової анотації складається з 3862 сутностей графу знань, включаючи як грубі, так і дрібні сутності, які були напівавтоматично створено та перевірені вручну за рейтингами для візуального розпізнавання. Кожна сутність має щонайменше 200 відповідних відеоприкладів. Groud truth для кожного відео визначені системою анотацій відео YouTube на основі вмісту, метаданих, контекстуальних та користувацьких сигналів, є основними темами кожного відео.

Ключові слова: *інтелектуальний аналіз даних, YouTube, програмне забезпечення, технології програмування Python, машинне навчання, відеохостинг, відеоаналітика, YouTube Data API, YouTube Analytics, метрики відеоаналітики, візуалізація даних.*

Постановка проблеми. YouTube є найбільш популярною платформою для завантаження, перегляду та спільного використання відеоконтенту. Кожної хвилини на YouTube завантажується понад сотні годин відео, що створює величезний обсяг даних, які можна проаналізувати для розуміння трендів, виявлення популярного контенту та виявлення нових можливостей для залучення аудиторії. Актуальність даного дослідження полягає в постійному зростанні важливості цієї платформи як основного джерела відеоконтенту та інформації для користувачів у всьому світі. З розширенням обсягів відео та аудиторії YouTube стає все важливіше дослідження та аналіз великих обсягів даних, щоб зрозуміти тенденції, попит, споживчі звички та ефективність контенту. Розробка програмного забезпечення, спрямованого на аналіз даних YouTube, відповідає на потребу в інструментах для отримання цінної інформації з цієї платформи для різноманітних цілей, включаючи маркетинг, дослідження громадської думки, розвиток контенту та аналіз трендів.

Аналіз останніх досліджень і публікацій. *YouTube* – це велика відеоплатформа, що дозволяє

користувачам завантажувати, переглядати та ділитися відеоконтентом. Заснований у 2005 році і придбаний компанією Google у 2006 році, YouTube став найпопулярнішим сервісом відеохостингу у світі. YouTube містить різноманітний контент, від коротких відео-блогів до повнометражних фільмів, і використовується мільйонами користувачів щодня для розваг, навчання та спілкування. Огляд наукових статей про YouTube розкриває різноманітні аспекти відеоконтенту та соціальних мереж, що включають культурну участь, вплив та кореляцію в соціальних мережах, рекомендаційні системи та статистику відео. Наприклад, у статті [1] обговорюється роль платформи YouTube у створенні нової, спільної форми онлайн-культури та її вплив на ширші культурні та соціальні традиції. У статті Наамана, Боасе та Лая [2] аналізується вміст потоків соціальної обізнаності, які використовуються в комп'ютерній спільній роботі для підвищення обізнаності щодо конкретних тем. Автори стверджують, що зміст цих потоків стосується не лише теми, яка розглядається, але й людей, які беруть участь, та їхніх стосунків. Вони пропонують метод аналізу вмісту цих потоків, заснований на концепції «егоцентричного вбудовування», яка передбачає представлення людей, залучених у потік, як вузли на графі, а вміст потоку як межі між ними. Автори демонструють корисність цього методу, аналізуючи вибірковий потік соціальної обізнаності та порівнюючи результати з більш простим аналізом на основі частоти ключових слів.

У роботі [3] досліджуються способи зв'язку впливу та кореляції в соціальних мережах і надається детальний аналіз теми. Автори використовують дані з різних джерел, щоб підтвердити свої висновки та представити свою роботу в чіткій і стислій формі. Загалом ця стаття є важливим внеском у сферу аналізу даних соціальних мереж і дає цінну інформацію про стосунки між різними особами та групами в онлайн-спільнотах.

У статті [4] представлено метод об'єднання конкуруючих думок із соціальних мереж для покращення рекомендацій фільмів. Метод передбачає агрегування думок кількох користувачів і їх використання для створення рекомендацій. У роботі наведені експерименти та результати, що свідчать про ефективність запропонованого методу.

У роботі [5] представлено систему рекомендацій для YouTube, який є вебсайтом для обміну відео. Система рекомендацій базується на підході колаборативної фільтрації, що означає, що вона рекомендує відео користувачам на основі шаблонів відео, які переглядали схожі користувачі. У статті описано модель та оцінку системи рекомендацій, яка використовувалася для рекомендації відео користувачам YouTube.

Автори статті [6] проаналізували роль систем рекомендацій у збільшенні переглядів відео на YouTube. Вони виявили, що система рекомендацій мала значний вплив: 10% найпопулярніших відео в їх наборі даних отримали 62% від загальної кількості переглядів. Дослідження показало, як можна оптимізувати алгоритми рекомендацій, щоб покращити залученість на таких платформах, як YouTube.

Автори статті [7] аналізують структуру інфраструктури YouTube, включаючи систему зворотного зв'язку з користувачами, яка дозволяє глядачам взаємодіяти з відео, надавати оцінки та коментарі. Вони вважають, що ця система зворотного зв'язку має вирішальне значення для розуміння соціальної динаміки платформи та надання рекомендацій користувачам. У документі представлено кілька прикладів, щоб проілюструвати різні способи взаємодії користувачів на YouTube і як це впливає на типи вмісту, який завантажується та рекомендований.

У роботі [8] досліджено використання теорії графів для аналізу соціальних мереж і поведінки перегляду користувачів YouTube. Автори зібрали дані про взаємодію користувачів YouTube і використали теорію графів для моделювання соціальних мереж і вимірювання центральності. У документі підкреслюється важливість центральності для розуміння соціальних мереж на YouTube і представлено кілька показників центральності, які використовуються в аналізі. Дослідження дає цінну інформацію про структуру та динаміку соціальних мереж на YouTube і про те, як вони впливають на поведінку перегляду.

Постановка завдання. Метою статті є розробка програмного забезпечення, яке дозволить виконувати статистичний аналіз даних про відео, виділяти тематики відео та виконувати інтелектуальний аналіз даних, який може бути корисним для аналітиків, маркетологів та створювачам контенту, а також для трансферного навчання та підходів до адаптації домену для відео.

Виклад основного матеріалу. Наявні програмні інструменти для аналізу даних платформи YouTube надають широкі можливості для розуміння та вивчення відеоконтенту, користувацьких взаємодій та трендів у споживанні контенту.

YouTube Data API – це інтерфейс програмування застосунків від YouTube, який надає доступ до різноманітної інформації про відео, канали, коментарі, статистику та інше. З його допомогою можна отримати доступ до публічної інформації та використовувати її для аналізу та обробки даних.

YouTube Analytics – це інструмент від YouTube, який надає детальну аналітику щодо відеоконтенту та каналів. YouTube Analytics дозволяє аналізувати кількість переглядів, час перегляду, реакції глядачів, заробіток та інші метрики для кращого розуміння ефективності контенту.

Social Blade – це зовнішній сервіс, який надає аналіз статистики каналів YouTube, включаючи підписників, перегляди, рейтинги та інші метрики. Social Blade також надає інструменти для порівняння каналів та вивчення трендів у споживанні контенту.

Tubular Labs – це інструмент для аналізу відеоконтенту на YouTube та інших платформах соціальних медіа. Tubular Labs надає розширену аналітику щодо залученості аудиторії, впливу відео та рекламних кампаній, а також допомагає знаходити та вивчати впливових творців контенту.

vidIQ – це розширення для веббраузера, яке надає розширену аналітику для каналів та відео на YouTube. vidIQ допомагає аналізувати ключові слова, рекомендації для покращення контенту, а також надає звіти про конкурентів та аудиторію.

Деякі програми спеціально розроблені для аналізу вмісту відео та аудіофайлів, такі як розпізнавання облич, аналіз настрою, виявлення об'єктів та інші. Інші програми спеціалізуються на виявленні трендів у відеоконтенті та ключових словах. Розглянуті програмні інструменти надають різноманітні можливості для аналізу даних платформи YouTube, що дозволяє власникам каналів, маркетологам та дослідникам приймати обгрунтовані рішення на основі даних.

Незважаючи на широку функціональність, розглянуті програмні інструменти для аналізу даних платформи YouTube також мають свої недоліки. Один з них полягає у обмеженому доступі до даних через API YouTube або обмеженнях, накладених самою платформою. Деякі інструменти можуть працювати повільно або недостатньо ефективно при обробці великих обсягів даних. Крім того, аналітичні дані можуть бути неповними або недостатньо точними, що може призвести до неточних висновків. Можливі шляхи вдосконалення включають розширення функціональності API та забезпечення більшого доступу до даних, оптимізацію алгоритмів обробки великих обсягів даних та розробку алгоритмів для підвищення точності аналітики та забезпечення достовірних результатів.

Аналіз даних сервісу YouTube за допомогою технологій програмування Python

Доступ до програмного забезпечення здійснюється з локального комп'ютера або віддаленого сервера з будь-якою ОС. Для роботи з даними використовується мова програмування Python 3.6 та Jupyter Notebook. Тому вони мають бути встановлені та налаштовані на комп'ютері. У першу чергу, щоб налаштувати середовище для аналізу даних, потрібно перейти до локального середовища програмування або середовища програмування на основі сервера. Для коректної роботи програмного забезпечення мають бути встановлені додаткові пакети, такі як pandas, numpy, matplotlib, seaborn, tensorflow. Тому виконуємо таку команду в терміналі:

```
pip install pandas numpy tensorflow matplotlib
seaborn
```

Далі запускаємо Jupyter Notebook у теці з програмним забезпеченням:

```
jupyter notebook
```

Потім ми виконуємо код, який містить файл youtube_analyser.ipynb. Для дослідження даних відео-сервісу Youtube було обрано датасет на сервісі відкритих даних Kaggle, обсягом 500Мб, з різними даними про понад 8 мільйонів відео, що розміщено на сервісі. YouTube-8M – це масштабний датасет відеоматеріалів із міткою, який складається з мільйонів ідентифікаторів відео YouTube, з високоякісними примітками із словника з 3800 візуальних сутностей. Він постачається з попередньо обчисленими аудіовізуальними функціями з мільярдів кадрів та аудіо-сегментів, розроблених для розміщення на одному жорсткому диску. Це дає можливість тренувати потужну базову модель на цьому наборі даних менше ніж за день на одному графічному процесорі. У той же час, масштаб і різноманітність набору даних можуть дати можливість глибокому вивченню складних аудіовізуальних моделей, які можуть тривати тижні, щоб навчитися навіть розподіленим способом.

Відеозаписи відбираються рівномірно, щоб зберегти різноманітне розповсюдження популярного вмісту на YouTube з урахуванням кількох

обмежень, вибраних для забезпечення якості та стабільності набору даних:

- кожне відео має бути загальнодоступним і мати не менше 1000 переглядів;
- кожне відео має тривати від 120 до 500 секунд;
- кожне відео має бути пов'язане щонайменше з однією сутністю з цільової лексики (класом);
- вміст для дорослих та чутливий видаляється (як визначено автоматизованими класифікаторами).

Набір даних представляє понад 350 000 годин відео і зазвичай потребує сотень терабайт пам'яті. Для оброблення цього набору даних (з обробкою відео в режимі реального часу на один процесор) знадобиться також 50-ти річне обчислення процесорами. Для зменшення місць зберігання та обчислювальних місць використовуються заздалегідь обчислені та стислі функції, які дають змогу тренувати модель на цьому наборі даних менше, ніж за день, на одному графічному процесорі. Відеоролики попередньо було оброблено для отримання 1.3 мільярдів найсучасніших візуальних функцій та 1.3 мільярдів аудіофункцій. Ми витягуємо функції на рівні відео, а також функції на рівні кадрів та сегментів (при роздільній здатності 1 секунди). Візуальні особливості були вилучені за допомогою моделі анотацій зображення Inception-V3, підготовленої на ImageNet. Звукові функції були витягнуті за допомогою звукової моделі VGG. І візуальні, і аудіофункції були оброблені алгоритмом PCA та квантовані для розміщення на одному жорсткому диску. Комбінований набір усіх функцій має розмір менше 2 ТБ.

Лексика цільової анотації складається з 3862 сутностей графу знань, включаючи як грубі, так і дрібні сутності, які були напівавтоматично створені та перевірені вручну за рейтингами, щоб бути візуально впізнаваними. Кожна сутність має щонайменше 200 відповідних відеоприкладів. Суб'єкти об'єднані в 24 вертикалі на високому рівні (рис. 1).

Groud truth для кожного відео, визначені системою анотацій відео YouTube на основі вмісту, метаданих, контекстуальних та користувацьких сигналів, є основними темами кожного відео. Кількість міток Groud truth на відео коливається від 1 до 23, в середньому 3,01 на відео.

У кожного відео є:

- id: унікальний ідентифікатор для відео, при тренуванні – це відео-ідентифікатор YouTube, а в тесті / валідації ці дані є анонімізовані;
- мітки: список міток цього відео;
- кожен кадр має RGB: float масив довжиною 1024;
- кожен кадр має аудіо: float масив довжиною 128.

У більшості пошукових запитів в мережі Інтернет пошук і ранжування відео виконується

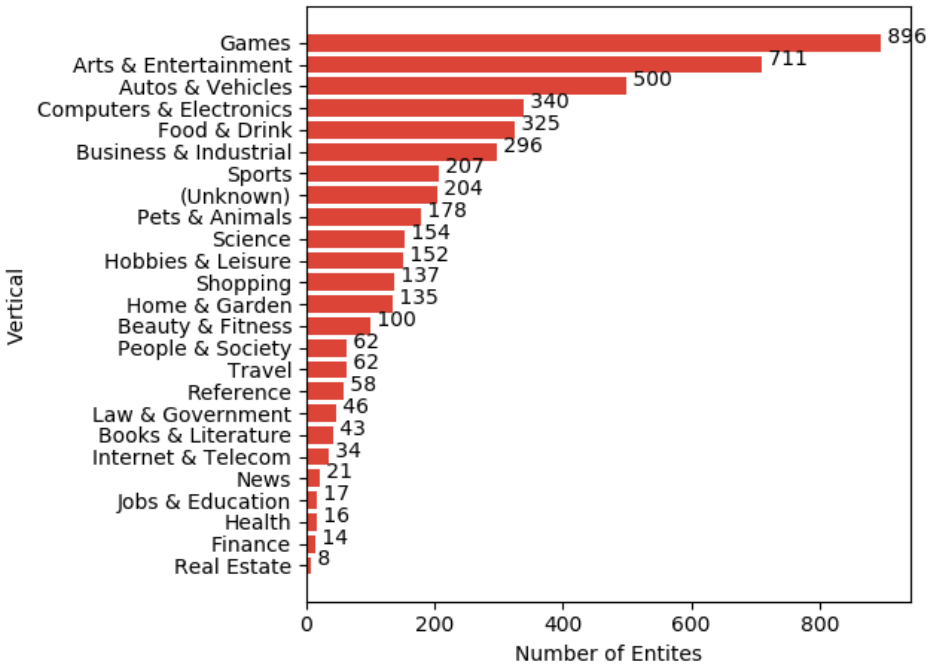

Рис. 1. Розподіл кількості сутностей за тематиками у відео

шляхом узгодження термінів запиту з метаданими та іншими сигналами рівня відео. Однак ми знаємо, що відео може містити теми, які не завжди є бажаними для користувача, наприклад, у відео з назвою та метаданими, що відповідають розважальним відео чи відео з тваринами, може зустрітись зовсім інший контент усередині, адже створювачі контенту можуть навмисно підлаштовувати метадані для більшого обсягу переглядів. Локалізація тематик на основі не метаданих, а контенту самого відео може включати такі програми, як покращений пошук відео, узагальнення відео та виділення основних моментів, виявлення моментів дії, що допоможе у поліпшенні безпеки відеоконтенту та багатьох інших задачах. У разі успіху цієї місії Youtube, нові моделі машинного навчання значно покращать розуміння відео для всіх, не лише визначаючи теми, що стосуються відео, а й визначатимуть, де у відео вони з'являються, що може як економити час користувачів, так і захистити їх від фейкового та нерелевантного контенту.

Створене програмне забезпечення міститься у Jupyter Notebook середовищі, програмне забезпечення подається у вигляді окремих функціональних блоків, кожен з яких містить невелику кількість програмного коду, та відповідає за виконання певного обмеженого набору функцій з завантаження та підготовки даних, а також їх подальшого аналізу різними методами. Головна особливість такої побудови програмного забезпечення є його доступність, та гнучкі можливості з тестування окремих блоків, та «нелінійного» виконання. А саме, мається на увазі те, що код не компілюється у єдину монолітну структуру програмного забезпечення, яка має початок (зазвичай це функція main), і кінець роботи. У IPython-ноутбуці є можливим почленне виконання блоків, проведення одних і тих самих операцій декілька разів, інплейс тестування створених методів та багато іншого. Саме такі можливості є важливими при роботі з даними, адже завжди необхідно переглядати проміжні результати, виводити дані, щоб впевнитись у їх коректності. Навчання моделей може займати багато часу, і при невірних чи непротестованих модулях, об'єднувати їх у один чи декілька файлів може завдати значних втрат у часі на виконання некоректного коду.

Розроблене програмне забезпечення містить наступні модулі.

Модуль завантаження даних та бібліотек відповідає за підключення усіх необхідних бібліотек для машинного навчання, аналізу даних, побудови графіків тощо. Також цей модуль завантажує дані з таблиць, та фреймів бібліотеки TensorFlow для подальшого опрацювання. Нижче наведено фрагмент коду, який здійснює імпорт необхідних бібліотек (рис. 2).

Далі ми опрацьовуємо файли тестового датасету, завантажуємо зміст файлів у пам'ять (рис. 3).

Далі ми зчитуємо таблицю сутностей (рис. 4).

Модуль візуалізації даних виконує побудову стовпчикових діаграм, графів зв'язку та іншої інформації.

На рис. 5 виведено 30 найбільш поширених тематик сутностей у представленому датасеті.

Оперуючи даними з таблиці, ми можемо побудувати графіки для нашого дослідження, інфор-

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import matplotlib.pyplot as plt
import csv
import networkx as nx
from subprocess import check_output
from wordcloud import WordCloud, STOPWORDS
import tensorflow as tf
from IPython.display import YouTubeVideo
plt.style.use('ggplot')

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory

import os
print(os.listdir("../input"))

import warnings
warnings.filterwarnings('ignore')
```

Рис. 2. Імпорт необхідних бібліотек

```
vid_ids = []
labels = []

for example in tf.python_io.tf_record_iterator(frame_lvl_record):
    tf_example = tf.train.Example.FromString(example)
    vid_ids.append(tf_example.features.feature['id']
                   .bytes_list.value[0].decode(encoding='UTF-8'))
    labels.append(tf_example.features.feature['labels'].int64_list.value)
```

```
print('Number of videos in this tfrecord: ',len(vid_ids))
print ('Number of labels in this tfrecord: ', len (labels))
print('Picking a youtube video id:',vid_ids[15])
```

```
Number of videos in this tfrecord:  1015
Number of labels in this tfrecord:  1015
Picking a youtube video id: FF00
```

Рис. 3. Зчитування TFrecord файлу

```
vocabulary = pd.read_csv('./vocabulary.csv')
vocabulary.head()
```

	Index	TrainVideoCount	KnowledgeGraphId	Name	WikiUrl	Vertical1	Vertical2	Vertical3	WikiDescription
0	0	788288	/m/03bt1gh	Game	https://en.wikipedia.org/wiki/Game	Games	NaN	NaN	A game is structured form of play, usually und...
1	1	539945	/m/01mw1	Video game	https://en.wikipedia.org/wiki/Video_game	Games	NaN	NaN	A video game is an electronic game that involv...
2	2	415890	/m/07yv9	Vehicle	https://en.wikipedia.org/wiki/Vehicle	Autos & Vehicles	NaN	NaN	A vehicle is a mobile machine that transports ...
3	3	378135	/m/01jddz	Concert	https://en.wikipedia.org/wiki/Concert	Arts & Entertainment	NaN	NaN	A concert is a live music performance in front...
4	4	286532	/m/09jwl	Musician	https://en.wikipedia.org/wiki/Musician	Arts & Entertainment	NaN	NaN	A musician is a person who plays a musical ins...

```
vocabulary.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
Index               1000 non-null int64
TrainVideoCount     1000 non-null int64
KnowledgeGraphId    1000 non-null object
Name                988 non-null object
WikiUrl             988 non-null object
Vertical1           1000 non-null object
Vertical2           153 non-null object
Vertical3           12 non-null object
WikiDescription     988 non-null object
dtypes: int64(2), object(7)
memory usage: 70.4+ KB
```

Рис. 4. Відображення інформації з таблиці сутностей

мація з яких може бути доволі корисною, враховуючи те, що було проведено попередні перевірки якості та актуальності датасету. На рис. 6 зображено граф зв'язків різних сутностей, в залежності від того, наскільки часто відео з спільними тематиками зустрічались у даних.

Модуль аналізу даних виконує аналіз даних за різними критеріями, у ньому містяться імплементації методів машинного навчання та статистичного аналізу даних, візуалізація результатів дослідження.
У цьому модулі представлено реалізації методів лінійної регресії, підрахунку кореляцій між змінними та інші методи статистичного аналізу даних.

Спочатку було проаналізовано вміст датасету, кількість його елементів, тощо (рис. 7).

Далі було проаналізовано розподіл різних метрик, побудовано графіки (рис. 8), та підраховано квантилі метрик. Також було порівняно категорії відео за мірою їх наявності у датасеті, проаналізовано розподіл переглядів між категоріями, те саме було проаналізовано для лайків, дизлайків та коментарів під відео.

Було проведено аналіз інтересу до відео певних категорій, в залежності від кількості лайків, дизлайків та коментарів (рис. 9) і було отримано цікаві результати дослідження, а саме:

- коментарі найбільше впливають на інтерес до відео у категоріях How To, Peoples & Blogs, Entertainment;

```
In [31]: labels_count_dict = dict(top_n)
         labels_count_df = pd.DataFrame.from_dict(labels_count_dict, orient='index').reset_index()
         labels_count_df.columns = ['label', 'count']
         labels_count_df['label'] = labels_count_df['label'].map(label_mapping, na_action='ignore')
         TOP_labels = list(labels_count_df['label'])[:n]
         fig, ax = plt.subplots(figsize=(10,7))
         sns.barplot(y='label', x='count', data=labels_count_df)
         plt.title('Top {} labels with sample count'.format(n))

Out[31]: Text(0.5, 1.0, 'Top 30 labels with sample count')
```

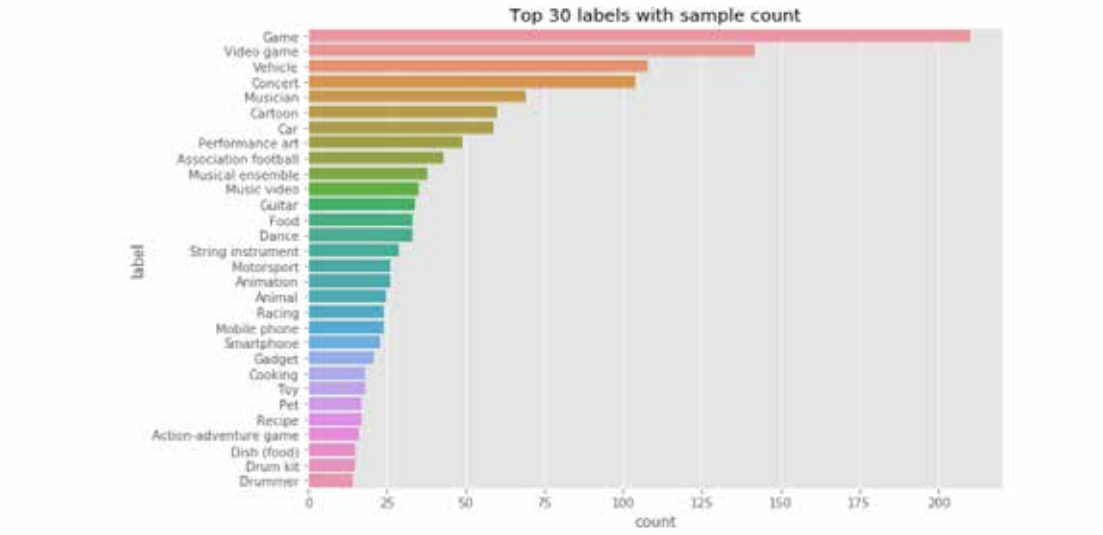

Рис. 5. Діаграма популярних тематик сутностей

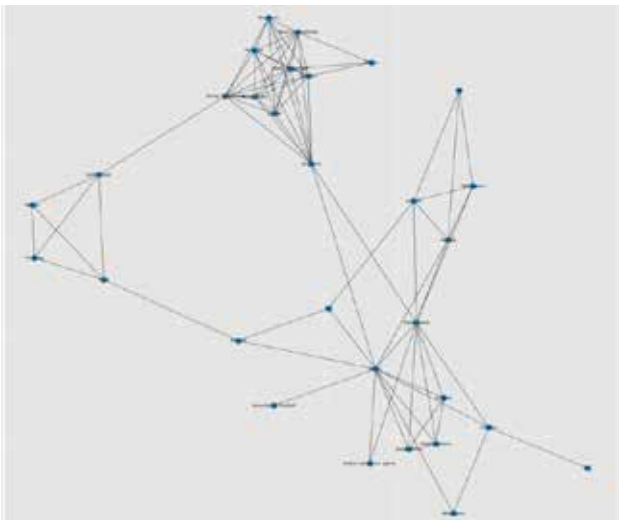

Рис. 6. Граф зв'язків основних тематик у датасеті

```
In [4]: df_yout = pd.read_csv("./USvideos.csv")

In [5]: #Looking some information of the data
        print(df_yout.shape)
        print(df_yout.nunique())

(40949, 16)
video_id                  6351
trending_date              205
title                     6455
channel_title             2207
category_id                 16
publish_time              6269
tags                      6055
views                    40478
likes                    29850
dislikes                  8516
comment_count            13773
thumbnail_link            6352
comments_disabled            2
ratings_disabled             2
video_error_or_removed       2
description               6901
dtype: int64
```

Рис. 7. Загальна інформація про датасет

- дизлайки очікувано впливають на заволікання людей до відео з категорії Politics;
- кількість лайків дуже важлива для категорії «Музика», про що варто замислитися музикантам початківцям, лейблам, орієнтованим на США.

Також у рамках дослідження було побудовано матрицю кореляцій для параметрів, щоб отримати відносну оцінку їх зв'язку один з одним. З отриманої матриці (рис. 10) випливає, що лайки та дизлайки дуже слабо пов'язані, що є логічно, адже різні люди за різними критеріями незалежно одне від одного ставлять їх, також видно, що кількість коментарів трохи більше пов'язана з лайками, ніж з дизлайками, проте зв'язок все одно не дуже сильний. Також було проведено додаткові аналізи тегів та згенеровано діаграми ключових слів, що зустрічається у тегах та назвах.

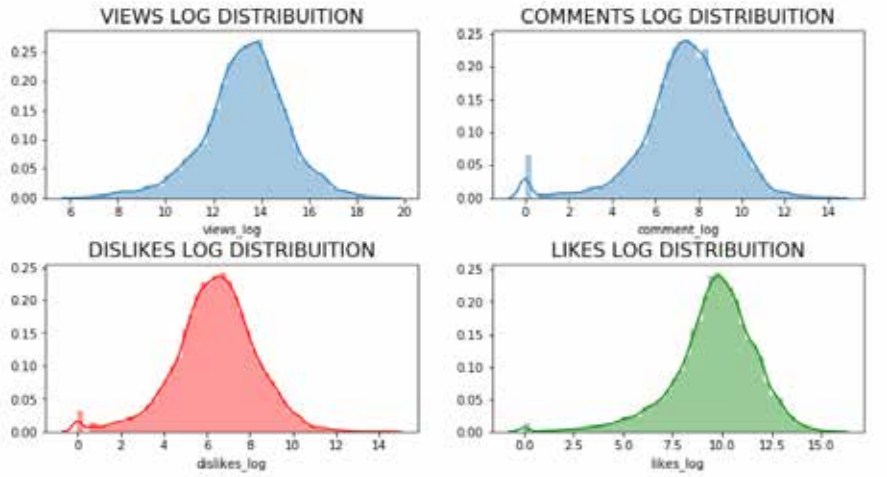

Рис. 8. Розподіл значень метрик відео

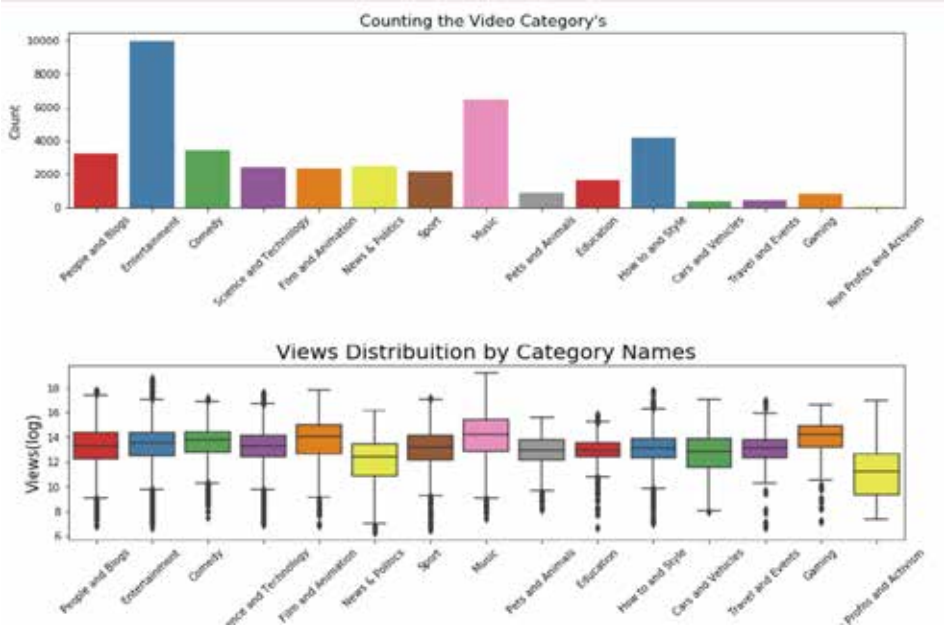

Рис. 9. Кількість представлених відео по категоріям, розподіл переглядів між категоріями

За допомогою бібліотек мови Python є можливим реалізація поглибленого аналізу, візуалізації даних та машинного навчання. При цьому завдяки простоті та інтуітивності мови та її синтаксису, для виконання цих складних операцій не витрачається багато часу на саме програмування, а більше часу приділяється самому аналізу та концепції (рис. 11).

При аналізі інтересу до відео в залежності від обраних метрик, ми ввели до даних нові змінні, а саме відношення обраної метрики до загальної кількості переглядів, що вимірюється у відсотках (рис. 12).

Ця метрика показує, яка кількість користувачів, з тих, хто переглянув відео, потім зробив ту чи іншу дію по відношенню до цього відео (лайк, дизлайк, коментар). Код до цієї частини аналізу наведено нижче (рис. 13).

Висновки. У даному дослідженні створено програмне забезпечення для операцій відкриття, зчитування, інтерпретації та візуалізації даних платформи YouTube. Програмне забезпечення подається у вигляді окремих функціональних блоків, кожен з яких містить невелику кількість програмного коду, та відповідає за виконання певного обмеженого набору функцій з завантаження та підготовки даних, а також їх подальшого аналізу різними методами. Реалізовані методи для інтелектуального аналізу даних, такі як лінійна регресія, розрахунок кореляцій, агрегація табличних даних.

```
In [20]: plt.figure(figsize = (10,8))

         #Let's verify the correlation of each value
         sns.heatmap(df_yout[['like_rate', 'dislike_rate', 'comment_rate', 'comment_log',
                'views_log','likes_log','dislikes_log', "category_name"]].corr(), annot=True)
         plt.show()
```

Рис. 10. Матриця кореляцій метрик відео

```
print("Category Name count")
print(df_yout.category_name.value_counts()[:5])

plt.figure(figsize = (14,9))

plt.subplot(211)
g = sns.countplot('category_name', data=df_yout, palette="Set1")
g.set_xticklabels(g.get_xticklabels(),rotation=45)
g.set_title("Counting the Video Category's ", fontsize=15)
g.set_xlabel("", fontsize=12)
g.set_ylabel("Count", fontsize=12)

plt.subplot(212)
g1 = sns.boxplot(x='category_name', y='views_log', data=df_yout, palette="Set1")
g1.set_xticklabels(g.get_xticklabels(),rotation=45)
g1.set_title("Views Distribuition by Category Names", fontsize=20)
g1.set_xlabel("", fontsize=15)
g1.set_ylabel("Views(log)", fontsize=15)

plt.subplots_adjust(hspace = 0.9, top = 0.9)

plt.show()
```

Рис. 11. Аналіз категорій та побудова графіків за допомогою Python

Одним з подальших напрямків роботи може бути поглиблене дослідження попиту та споживчих звичок аудиторії YouTube, включаючи вивчення впливу різних факторів на перегляди та взаємодію з контентом. Крім того, можна дослідити розвиток нових технологій штучного інтелекту для отримання більш глибокого розуміння тенденцій та структури відеоконтенту на YouTube. Такі дослідження можуть допомогти покращити якість контенту, зрозуміти потреби аудиторії та оптимізувати стратегії маркетингу та розвитку каналів на цій платформі.

```
df_yout['likes_log'] = np.log(df_yout['likes'] + 1)
df_yout['views_log'] = np.log(df_yout['views'] + 1)
df_yout['dislikes_log'] = np.log(df_yout['dislikes'] + 1)
df_yout['comment_log'] = np.log(df_yout['comment_count'] + 1)

plt.figure(figsize = (12,6))

plt.subplot(221)
g1 = sns.distplot(df_yout['views_log'])
g1.set_title("VIEWS LOG DISTRIBUITION", fontsize=16)

plt.subplot(224)
g2 = sns.distplot(df_yout['likes_log'],color='green')
g2.set_title('LIKES LOG DISTRIBUITION', fontsize=16)

plt.subplot(223)
g3 = sns.distplot(df_yout['dislikes_log'], color='r')
g3.set_title("DISLIKES LOG DISTRIBUITION", fontsize=16)

plt.subplot(222)
g4 = sns.distplot(df_yout['comment_log'])
g4.set_title("COMMENTS LOG DISTRIBUITION", fontsize=16)

plt.subplots_adjust(wspace = 0.2, hspace = 0.4,top = 0.9)

plt.show()
```

Рис. 12. Аналіз розподілів метрик відео

```
plt.figure(figsize = (14,6))

g = sns.boxplot(x='category_name', y='likes_log', data=df_yout, palette="Set1")
g.set_xticklabels(g.get_xticklabels(),rotation=45)
g.set_title("Likes Distribuition by Category Names ", fontsize=15)
g.set_xlabel("", fontsize=12)
g.set_ylabel("Likes(log)", fontsize=12)
plt.show()
```

а)

```
plt.figure(figsize = (14,6))

g = sns.boxplot(x='category_name', y='dislikes_log', data=df_yout, palette="Set1")
g.set_xticklabels(g.get_xticklabels(),rotation=45)
g.set_title("Dislikes distribuition by Category's", fontsize=15)
g.set_xlabel("", fontsize=12)
g.set_ylabel("Dislikes(log)", fontsize=12)
plt.show()
```

б)

```
plt.figure(figsize = (14,6))

g = sns.boxplot(x='category_name', y='comment_log', data=df_yout, palette="Set1")
g.set_xticklabels(g.get_xticklabels(),rotation=45)
g.set_title("Comments Distribuition by Category Names", fontsize=15)
g.set_xlabel("", fontsize=12)
g.set_ylabel("Comments Count(log)", fontsize=12)

plt.show()
```

в)

Рис. 13. Фрагменти коду для роботи з новими метриками

Список літератури:

1. Burgess J. YouTube: Online video and participatory culture. *International journal of cultural studies*. 2008. 11(1). pp. 31-44. DOI: 10.1177/1367877907089475.

2. Naaman M., Boase J. and Lai C.H. Is it really about me? Message content in social awareness streams. *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. 2010. pp. 189-192. DOI: 10.1145/1718918.1718957.

3. Yang K., Kunegis J., Lommatzsch A. and Staab S. Influence and correlation in social networks. *Proceedings of the 20th international conference on World wide web*. 2011. pp. 7-8. DOI: 10.1145/1963405.1963506.

4. Zhang L., Liu B. and Zhang Y. Aggregating competing opinions from social networks for movie recommendation. *Proceedings of the 23rd international conference on Computational Linguistics*. 2010. pp. 1224-1232. DOI: 10.5555/1858681.1858817.

5. Davidson J., Liebald B., Liu J., Nandy P. and Van Vleet T. The YouTube video recommendation system. *Proceedings of the fourth ACM conference on Recommender systems*. 2010. pp. 293-296. DOI: 10.1145/1864708.1864770.

6. Zhou Renjie et al. The impact of YouTube recommendation system on video views. *ACM/SIGCOMM Internet Measurement Conference*. 2010. pp. 404-410. DOI: 10.1145/1879141.1879193.

7. Wattenhofer M., Wattenhofer R., & Zhu Z. The YouTube Social Network. *Proceedings of the International AAAI Conference on Web and Social Media*. 2021. 6(1), pp. 354-361. https://doi.org/10.1609/icwsm.v6i1.14243.

8. Cheng Xu et al. Statistics and Social Network of YouTube Videos. *2008 16th Interntional Workshop on Quality of Service*. 2008. pp. 229-238.

Oleshchenko L.M. YOUTUBE PLATFORM DATA ANALYSIS SOFTWARE

This article discusses the software tools designed to analyze data from the YouTube platform, provides an overview of their functionality and main capabilities. Software has been developed that allows to perform statistical analysis of large volumes of video data, highlight video themes and make intelligent data analysis that can be useful for analysts, marketers and content creators. The graphical interface of the software was developed using Rython programming language tools, the Jupyter-Notebook environment, NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, TensorFlow libraries were used. The implemented software system allows to process data in the form of csv tables, first of all, from the Kaggle service, from where the data was taken for research and data analysis.

As a result of the research, software was created for working with data, for the operations of opening, reading, interpreting and visualizing data. The software is provided in the form of separate functional blocks, each of which contains a small amount of software code, and is responsible for performing a certain limited set of functions for loading and preparing data, as well as their further analysis by various methods. Implemented methods for intelligent data analysis, such as linear regression, calculation of correlations, aggregation of tabular data.

The dataset consists of over 350,000 hours of video. Precomputed and compressed features are used to reduce storage and computation space, allowing the model to be trained on a dataset in less than a day on a single GPU. Videos have been pre-processed to obtain 1.3 billion state-of-the-art visual features and 1.3 billion audio features. Video-level features, as well as frame- and segment-level features (at 1 second resolution) were extracted. Visual features were extracted using the Inception-V3 image annotation model prepared on ImageNet. The sound features were extracted using the VGG sound model on the previous version of YouTube-8M. Both visual and audio features were PCA processed and quantized to fit on the same hard disk. The combined set of all features is less than 2 TB in size. The target annotation vocabulary consists of 3862 knowledge graph entities, including both coarse and fine entities, which were semi-automatically generated and manually checked against ratings for visual recognition. Each entity has at least 200 relevant video examples. Groud truth for each video is determined by YouTube's video annotation system based on content, metadata, contextual and user signals, are the main topics of each video.

Key words: *data mining, YouTube, software, Python programming technologies, machine learning, video hosting, video analytics, YouTube Data API, YouTube Analytics, video analytics metrics, data visualization.*