Assignment – 2

Shubham Kumar Choudhary

MT23093

Library used :

pandas: A powerful data manipulation and analysis library in Python, providing easy-to-use data structures and functions for working with structured data like tables and time series.

numpy: A fundamental package for scientific computing in Python, offering support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently.

ast: A Python built-in module used for abstract syntax tree (AST) manipulation, providing functions like ast.literal_eval() to safely evaluate Python expressions in strings containing literal structures such as lists and dictionaries.

requests: An elegant and simple HTTP library for Python, allowing easy sending of HTTP requests and handling of responses, making it ideal for web scraping, API interaction, and other HTTP-based tasks.

PIL (Python Imaging Library): A library for adding image processing capabilities to Python, enabling tasks like opening, manipulating, and saving many different image file formats.

tensorflow: An open-source machine learning framework developed by Google, offering comprehensive tools and libraries for building and training deep learning models across a variety of platforms and devices.

VGG16: A pre-trained convolutional neural network (CNN) architecture introduced by the Visual Geometry Group at the University of Oxford, commonly used for image classification tasks due to its simplicity and effectiveness.

image: A submodule of tensorflow.keras.preprocessing providing utilities for image preprocessing tasks such as loading, resizing, and augmenting images before feeding them into neural networks for training or inference.

preprocess_input: A function from the VGG16 module in tensorflow.keras.applications, used to preprocess input images according to the requirements of the VGG16 model before passing them for prediction or feature extraction.

normalize: A function from tensorflow.keras.utils, used to normalize arrays or tensors along specified axes, typically used to preprocess data before feeding into machine learning models to improve convergence and performance.

METHODOLOGY:

Q1:

Approach: Utilize a pre-trained Convolutional Neural Network (CNN) architecture (e.g., VGG16) to extract relevant features from images.

Methodologies: Apply basic image pre-processing techniques such as resizing, geometric orientation adjustments, random flips, brightness and exposure adjustments.

Assumptions: Assume the pre-trained CNN model has been trained on a large dataset like ImageNet and can effectively capture image features.

Results: Extracted features are normalized to ensure consistency and compatibility with subsequent processing steps.

Q2:

Approach: Implement text pre-processing techniques such as lower-casing, tokenization, punctuation removal, stop word removal, stemming, and lemmatization.

Methodologies: Calculate TF-IDF scores for textual reviews, which quantify the importance of words in documents relative to a corpus.

Assumptions: Assume TF-IDF scores effectively capture the relevance of words in reviews and contribute to similarity calculations.

Results: Extracted TF-IDF scores are saved using the pickle module for further analysis.

Q3:

Approach: Utilize cosine similarity to find the most similar images and reviews to the input image and review pair, respectively.

Methodologies: Compute cosine similarity between the input image features and all image features, as well as between the input review TF-IDF scores and all reviews' TF-IDF scores.

Assumptions: Assume cosine similarity provides a suitable measure for comparing image features and TF-IDF scores, and the top similar images and reviews are relevant to the input pair.

Results: Top similar images and reviews, along with their similarity scores, are saved using the pickle module.

Q4:

Approach: Calculate a composite similarity score by averaging the cosine similarity scores obtained from image and text retrieval.

Methodologies: Rank the pairs based on the composite similarity score to identify the most relevant (image, review) pairs.

Assumptions: Assume combining image and text retrieval results in a more comprehensive similarity measure, reflecting both visual and textual content.

Results: Top-ranked (image, review) pairs, along with their composite similarity scores, are presented and analyzed.