Name : Krishna Somani Roll No: 2021058

BDA Assignment 2

Code Explanation

Overview of functions:

1. preprocess_text(text, lemmatize=True, stem=False):
    a. This function cleans and prepares the input text for analysis.
    b. Regular Expression (re): Used to remove non-alphabetic characters and convert the text to lowercase.
    c. This function further utilizes “NLTK's stopwords” to remove common words that do not add significant meaning (e.g., "and", "the").
    d. Further it uses “WordNetLemmatizer” to reduce words to their base form (lemmatization) if the `lemmatize` parameter is set to True.
    e. If stem is True (argument inside function is provided as true), it applies PorterStemmer to further reduce words to their root forms (stemming).
    f. Finally it returns the cleaned text as a single string.

2. text_to_minhash(text, num_perm=128):
    a. Converts a given text into a “MinHash” object from the datasketch library.
    b. MinHash allows for efficient similarity estimation between sets by creating a compact representation of the input text.
    c. The function iterates through the words in the text, updating the MinHash object with each word encoded in UTF-8.
    d. Returns the MinHash object, which provides a way to compare the similarity of different texts later.

3. lsh_similarity(ids, texts, num_perm=128, threshold=0.4):
    a. This is the main function to find similar items based on the provided texts.
    b. MinHashLSH: An object from the datasketch library that enables locality-sensitive hashing, allowing for quick retrieval of similar items based on their MinHash representations.
    c. The function then preprocesses the texts using the preprocess_text function.
    d. Then it Initializes an LSH object with a specified threshold for similarity and the number of permutations for MinHash.
    e. It inserts each MinHash into the LSH.
    f. For each ID, it queries the LSH to find the top 5 similar items, ensuring the original item is excluded from its own results.
    g. Returns a dictionary of predictions containing IDs and their corresponding similar items.

4. evaluate_model(predictions, ground_truth_file):
    a. It evaluates the accuracy of the model's predictions against the ground truth data.
    b. Reads ground truth data from a JSON file (items.json)
    c. Calculates an intersection score for each predicted item compared to the actual items using set operations, which helps in determining how many of the predicted similar items are actually correct.
    d. Returns the average score and a list of individual scores.

Main Execution Flow

1. Reading Input: Reads IDs and texts from files and preprocesses the texts using the preprocess_text function.
2. Getting Predictions: Calls the lsh_similarity function to obtain similar items based on the provided texts and IDs
3. Evaluating Model: Uses the evaluate_model function to compute the average score of the predictions against the ground truth.
4. Output: Displays the average score of the model’s predictions.

Conclusion & Results

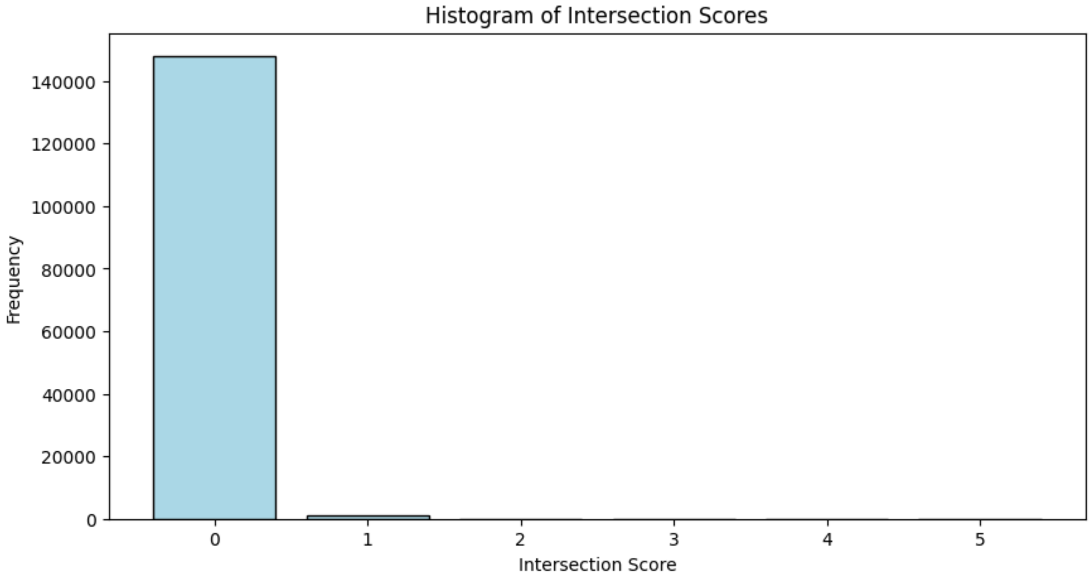
Overall process involved building an LSH model to retrieve the top 5 similar items based on text features. The model's performance was evaluated using the average similarity score, comparing retrieved items to the ground truth.

a) Average Similarity Score: The mean score is approximately 0.007, indicating low similarity across items.

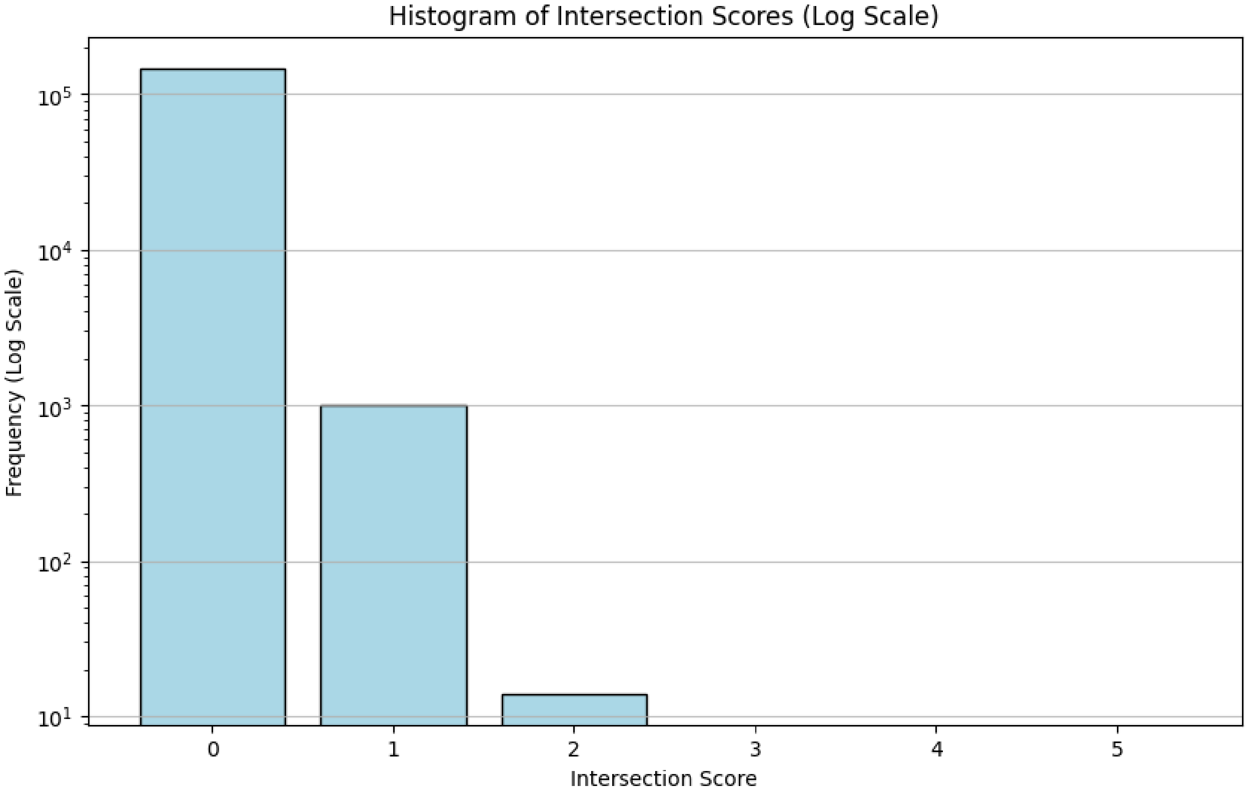
b) Reason for Low Score: This low score is due to minimal overlap in text content across items, which limits the LSH model's ability to identify meaningful similarities, as it relies on shared textual features.

In summary, LSH is best suited for datasets with some inherent textual similarity for effective similarity retrieval.

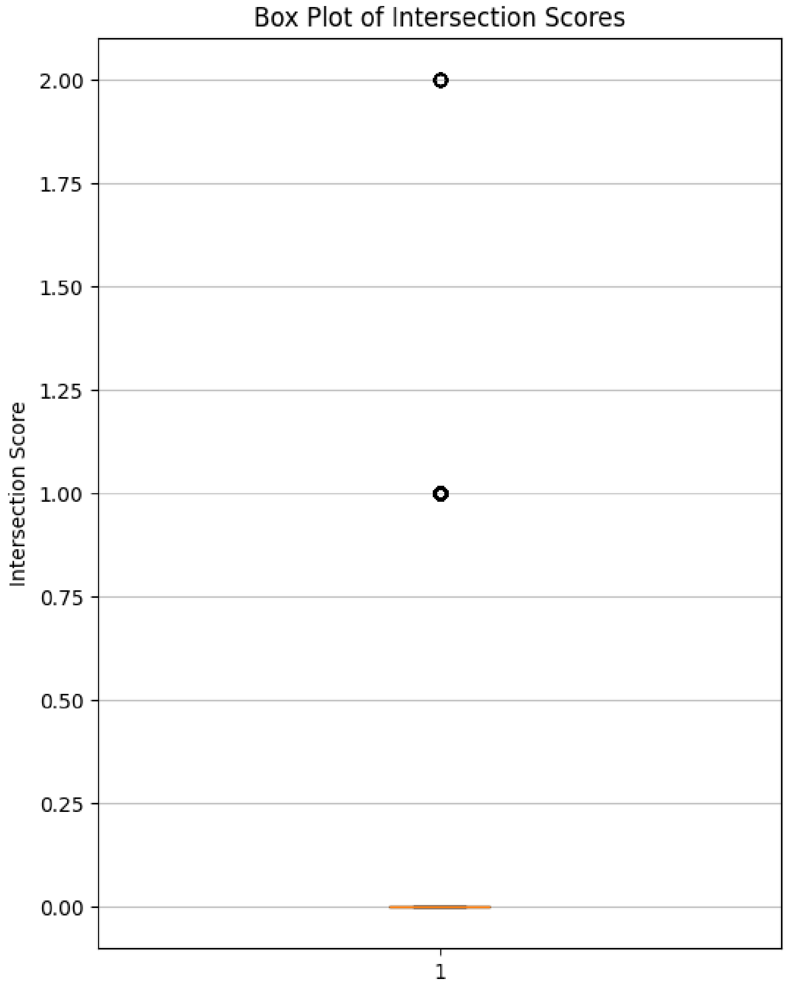
Plots



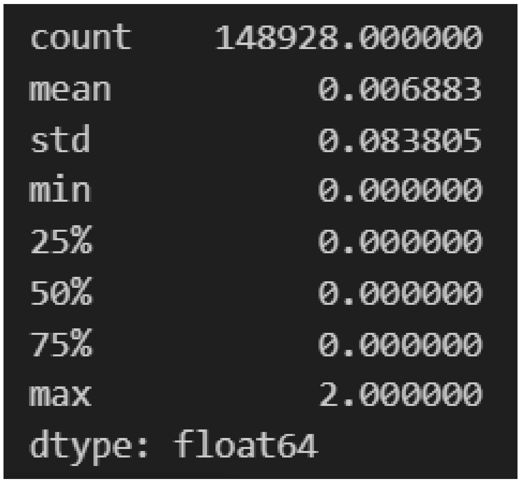
In logarithmic scale



Box Plot



pandas.describe()



```
count    148928.000000
mean          0.006883
std           0.083805
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max           2.000000
dtype: float64
```