Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2021, 8(7):79-83

Research Article ISSN: 2394 - 658X

Automated Object Deletion in Google Cloud Storage: Introducing the Clean-up-gcs-bucket Library

Preyaa Atri

Preyaa.atri91@gmail.com

ABSTRACT

Managing large datasets within Google Cloud Storage (GCS) buckets is a critical aspect of efficient data management, particularly in the context of AI development. This paper introduces the Clean-up-gcs-bucket Python library, a tool designed to streamline the deletion of objects based on user-defined criteria, thereby optimizing storage utilization and enhancing the AI model development process. We delve into the library's functionality, exploring its capabilities for filtering objects by folder path and substring, which allows for precise control over data deletion, ensuring that only outdated or irrelevant data is removed, thus maintaining the integrity and relevance of datasets used for AI model training. Additionally, we discuss the library's impact on data management efficiency, usage considerations, and potential applications in various domains. We provide recommendations for maximizing the library's effectiveness and propose areas for future development, aiming to further enhance its capabilities and adaptability to evolving data management needs

Keywords: Google Cloud Storage, Object Deletion, Python Library, Data Management, AI

INTRODUCTION

Cloud storage platforms like Google Cloud Storage (GCS) offer a scalable and cost-effective solution for storing vast amounts of data. However, as datasets grow, the need to effectively manage and remove unwanted objects becomes increasingly crucial. Manual deletion can be time-consuming and error-prone, particularly for large buckets containing numerous objects. The Clean-up-gcs-bucket library addresses this challenge by providing an automated approach to object deletion based on user-specified criteria. By allowing users to set specific criteria for deletion improves the efficiency and effectiveness of data management tasks within GCS (Greca & Kosta, 2020). Organizations utilizing Google Cloud Platform for data storage can take advantage of the tailored deletion capabilities provided by this library, ensuring the efficient removal of unnecessary data based on predefined rules and filters (Hua et al., 2008).

PROBLEM STATEMENT

Large GCS buckets often accumulate redundant, obsolete, or erroneous data over time. Manually identifying and deleting such objects can be a tedious and error-prone task. Traditional scripting approaches may require significant coding expertise and lack user-friendly interfaces. The Clean-up-gcs-bucket library fills this gap by offering a user-centric solution for efficient object deletion.

SOLUTION

The Clean-up-gcs-bucket library provides a single function, Clean-up-gcs-bucket(), that simplifies object deletion within GCS buckets. This function accepts three arguments:

- **bucket_name (str):** The name of the GCS bucket to be processed.

- **folder_path (str, optional):** An optional argument specifying a folder path prefix to restrict deletion to a specific subfolder within the bucket. It must include a trailing slash (/). If omitted, the function considers all objects in the bucket.
- **substring (str, optional):** Another optional argument that allows filtering objects based on the presence (case-insensitive) of a specified substring within the object name. If not provided, all objects matching the folder_path criteria (if specified) will be deleted.

The function leverages the google-cloud-storage library (assumed to be pre-installed) to establish a connection with GCS. It then retrieves the specified bucket and iterates through each object. The deletion logic hinges on the presence of the substring argument. If not provided, all objects matching the folder_path criteria are deleted. Conversely, if a substring is provided, only objects containing the substring and matching the folder_path (if specified) are deleted. The function meticulously tracks the number of deleted objects and culminates by printing a summary message detailing the deletion operation, including the bucket name, number of deleted objects, and the used criteria (folder path and/or substring, if applicable).

FUNCTIONALITY AND USAGE

The Clean-up-gcs-bucket library offers a user-friendly function, Clean-up-gcs-bucket(), designed to streamline object deletion within GCS buckets. This function takes three arguments:

- **bucket_name (str):** This mandatory argument specifies the name of the GCS bucket you intend to process. The library interacts with this bucket to identify and delete objects based on the provided criteria.
- **folder_path (str, optional):** This optional argument allows you to restrict object deletion to a specific subfolder within the bucket. The path must include a trailing slash (/). If omitted, the function considers all objects residing within the specified bucket. This functionality proves useful when you only want to clean up unwanted data within a particular subfolder, preserving objects in other parts of the bucket.
- **substring (str, optional):** Another optional argument that empowers you to filter objects based on the presence of a specific substring (case-insensitive) within the object's name. If you don't provide a substring, the function deletes all objects matching the specified folder_path (if provided). Conversely, including a substring ensures that only objects containing the substring and matching the folder_path (if specified) are deleted. This filtering capability offers greater precision in object deletion, allowing you to target specific data sets or file types.

INSTALLATION

Installing the Clean-up-gcs-bucket library is a straightforward process that leverages the pip package manager commonly used for Python library installation. Here's how to get started:

1. Open your terminal or command prompt.
2. Ensure you have pip installed. If not, refer to the official Python documentation for installation instructions.
3. Execute the following command in your terminal:

Bash

```
pip install Clean-up-gcs-bucket #installs Clean-up-gcs-bucket Library
```

This command instructs pip to download and install the Clean-up-gcs-bucket library from the Python Package Index (PyPI). Once the installation is complete, you can start using the library in your Python projects.

EXAMPLE USAGE

Here's a practical example demonstrating how to utilize the Clean-up-gcs-bucket library:

```
Python
from Clean-up-gcs-bucket import clean-up-gcs-bucket

# Replace with your information
bucket_name = "my-bucket"
folder_path = "data/old_reports/" # Optional, specify a subfolder
substring = "report_2023" # Optional, filter by substring (case-
insensitive)
clean-up-gcs-bucket(bucket_name, folder_path, substring)
```

In this example, the code snippet targets the GCS bucket named "my-bucket". It specifies a folder_path of "data/old_reports/", instructing the library to delete objects only within this subfolder. Additionally, the substring argument is set to "report_2023" (case-insensitive), ensuring that only objects containing this substring within their names are deleted from the specified folder. After execution, the library will delete matching objects, printing a summary message detailing the number of deleted objects, bucket name, and the used criteria (folder path and substring).

DEPENDENCIES AND CONSIDERATIONS

The Clean-up-gcs-bucket library relies on the google-cloud-storage library to interact with Google Cloud Storage. It is assumed that this dependency is pre-installed on your system. If not, you can install it using pip install google-cloud-storage before utilizing the Clean-up-gcs-bucket library.

Important Considerations:

- **Destructive Nature:** The Clean-up-gcs-bucket library permanently deletes objects. Ensure you have proper backups in place before using the library to avoid accidental data loss.
- **Security:** The current implementation likely relies on user-defined credentials or environment variables. Consider using more secure authentication mechanisms like service accounts for enhanced security.
- **Advanced Usage:** For more sophisticated object deletion workflows, explore the capabilities offered by the underlying google-cloud-storage library.

By understanding these dependencies and considerations, you can effectively leverage the Clean-up-gcs-bucket library for efficient object deletion within your GCS buckets.

USES AND IMPACT

The Clean-up-gcs-bucket library significantly impacts data management within Google Cloud Storage, particularly benefiting AI development in several ways:

- **Efficient AI Model Training:** By automating the removal of outdated or irrelevant data, the library ensures that AI models are trained on the most recent and relevant datasets, leading to improved model performance and accuracy.
- **Streamlined Data Preprocessing:** Removing unnecessary data simplifies the preprocessing pipeline for AI tasks, reducing computational overhead and accelerating the preparation of data for model training.
- **Cost Optimization for AI Workloads:** Efficient data management through automated deletion helps optimize storage costs associated with GCS buckets, allowing organizations to allocate resources more effectively for AI model development and deployment.
- **Enhanced Data Governance for AI:** The library's filtering capabilities enable precise control over data retention, ensuring compliance with data governance policies and regulations specific to AI applications.
- **Improved Collaboration in AI Projects:** By maintaining a clean and organized GCS environment, the library facilitates collaboration among AI researchers and developers, as they can easily access and share relevant datasets.

In addition to its impact on AI, the Clean-up-gcs-bucket library offers broader benefits for data management:

- **Regular Removal of Obsolete Data**: Automating the deletion of expired or unnecessary data helps maintain a well-organized GCS environment, improving data accessibility and reducing storage costs.
- **Compliance with Data Retention Policies**: The library aids in adhering to industry-specific data retention regulations by systematically removing data that exceeds specified retention periods.
- **Efficient Cleanup of Erroneous Data**: Unintentionally uploaded or incorrect datasets can be swiftly removed using the library's filtering options, ensuring data integrity.

Overall, the Clean-up-gcs-bucket library streamlines data management processes, enhances efficiency, and reduces manual effort associated with object deletion in GCS. Its capabilities extend beyond general data management to significantly benefit AI development by optimizing storage, ensuring data relevance, and facilitating collaboration.

SCOPE AND LIMITATIONS

The Clean-up-gcs-bucket library excels at deleting objects based on folder path and substring criteria within a single GCS bucket. However, it is essential to acknowledge its limitations:

- **Single Bucket Support:** The library currently operates on a single bucket at a time. For scenarios involving multiple buckets, users would need to execute the function iteratively for each bucket. This can become cumbersome for users managing a large number of GCS buckets. Future iterations of the library could incorporate functionality to process deletion tasks across multiple buckets, potentially with the ability to specify inclusion or exclusion criteria for specific buckets.
- **Permanence of Deletion:** The library permanently deletes objects. It lacks functionalities for restoring accidentally deleted objects. This highlights the importance of emphasizing to users the need to have proper data backups in place before utilizing the library. Integration with versioning features offered by GCS could potentially be explored to enable retrieval of older versions of deleted objects in future developments.
- **Granular Control Limitations:** While the library offers filtering based on folder path and substring, it lacks more granular control mechanisms for object selection. For instance, users cannot specify deletion based on object size, creation date, or custom metadata properties. Expanding the library's capabilities to incorporate such filtering criteria would provide users with more flexibility and precision in managing their GCS object deletions.
- **Security Considerations:** The current library implementation likely relies on user-defined credentials or environment variables to authenticate with GCS. It would be beneficial to explore incorporating more robust authentication mechanisms, such as service accounts, to enhance security and access control.

CONCLUSION

The Clean-up-gcs-bucket library presents a valuable tool for streamlining object deletion within Google Cloud Storage, with a particular emphasis on enhancing efficiency in AI development pipelines. By automating the removal of outdated or irrelevant data, the library ensures that AI models are trained on the most relevant and up-to-date datasets, leading to improved model performance and accuracy. While the current implementation focuses on core functionalities like filtering by folder path and substring, future enhancements could expand its capabilities to include more granular control over object deletion, support for multiple buckets, and potential integration with versioning features for data recovery. As the demand for efficient data management in AI continues to grow, this library positions itself as a valuable asset in optimizing storage utilization, ensuring data relevance, and ultimately accelerating the development and deployment of AI models.

REFERENCES

[1]. Google Cloud Platform. [Online]. Cloud Storage Documentation. Available: https://cloud.google.com/storage/docs

[2]. S. Greca and A. Kosta, "The impact of the google cloud to increase the performance for a use case in e-commerce platform", International Journal of Computer Applications, vol. 177, no. 35, p. 9-13, 2020. https://doi.org/10.5120/ijca2020919748

[3]. Y. Hua, B. Xiao, D. Feng, & B. Yu, "Bounded lsh for similarity search in peer-to-peer file systems", 2008 37th International Conference on Parallel Processing, 2008. https://doi.org/10.1109/icpp.2008.25

[4]. M. Barisits, F. Barreiro, T. Beermann, K. Bhatia, K. De, A. Dubreuilet al., "The data ocean project", EPJ Web of Conferences, vol. 214, p. 04020, 2019. https://doi.org/10.1051/epjconf/201921404020

[5]. S. Ahmad and M. Afza, "A review of assured data deletion mechanism in cloud computing", International Journal of Engineering &Amp; Technology, vol. 7, no. 4.5, p. 329, 2018. https://doi.org/10.14419/ijet.v7i4.5.20101