BONUS QUESTIONS

1. Why is storing cleaned data in Azure Blob Storage important for real-time pipelines?

Storing cleaned data in Blob Storage ensures that downstream systems (analytics, reporting, ML models) always work with consistent and reliable data. Since Blob Storage is scalable and accessible globally, multiple services can consume the same trusted dataset in real time without duplication or errors.

2. What's the difference between pipeline artifacts and Blob Storage uploads?

- **Pipeline artifacts** → Temporary outputs stored inside Azure DevOps. They are mainly used to pass files between pipeline stages or for developers to download results after a run.
- **Blob Storage uploads** → Persistent storage in Azure. Files remain available long term and can be accessed by any external system or service (Power BI, Synapse, Databricks, etc.).

3. How would you handle failures in file uploads in a production setup?

In production, you'd add retry logic and logging. For example, if an upload fails due to a network issue, the script should retry a few times before failing. Errors should be logged and sent to monitoring tools (like Azure Monitor or Application Insights). You might also use dead-letter queues or alerts so that a data engineer is notified immediately and can re-trigger the upload.