Mihir Patel

CS 499

Prof. Sherin Aly

July 23, 2025

This artifact is a Python-based ETL (Extract, Transform, Load) script that retrieves customer churn data from AWS S3, cleans and transforms it, and prepares it for further analysis or loading into a data warehouse. The code is divided into well-organized functions for downloading, cleaning, and displaying the data. It uses exception handling and basic logging while following key software design principles.

I chose this artifact because it shows my ability to build clean, modular Python code that is production-ready. It highlights key software development practices such as:

* Writing reusable and maintainable code with proper function separation
* Securely accessing cloud resources using environment variables
* Logging and error handling for reliability
* Preparing a pipeline that could be easily integrated into a real business workflow

It reflects real-world data engineering work and demonstrates my competency in designing robust, scalable software solutions.

* Replaced hardcoded AWS credentials with environment variables to enhance security
* Added try-except blocks to handle connection or download failures
* Added comments and docstrings to improve clarity
* Split transformation logic into a separate script to keep code clean
* Replaced hardcoded values with variables for flexibility
* Improved logging to support better debugging and observability
* Handled edge cases like division by zero during feature engineering

Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals (software engineering/design/database).

While working on this enhancement, I became more confident in applying software engineering best practices. Initially, my code was functional but not ready for production. By refactoring the structure, handling errors properly, and making the code configurable, I was able to make it much more robust.

One of the more difficult parts was handling S3 download failures gracefully and making sure I wasn't exposing sensitive credentials. I also learned how important it is to add logging for better monitoring, especially if the pipeline fails. These small changes significantly increased the code’s quality and maintainability.

Overall, this enhancement gave me practical experience building reliable data pipelines and improved my understanding of what good software engineering looks like.