

ETL Process Documentation: Data Transfer Project

1. Introduction

In this project, I developed an ETL (Extract, Transform, Load) process to transfer shopping behavior data using Azure Data Factory, SQL Server, and Power BI for visualization and analysis. This documentation outlines the steps I took throughout the project, from setting up the data factory to performing advanced analytics in Power BI.

2. Data Source

The primary data source for this project was a dataset containing customer shopping behavior, including variables such as Customer ID, Age, Gender, Purchase Amount (USD), Review Rating, and other relevant details.

3. Setting Up Azure Data Factory

3.1. Creating the Data Factory

- I created a new Azure Data Factory instance through the Azure Portal, which would serve as the main service for managing data pipelines.

3.2. Creating a Storage Account

- I set up an Azure Storage Account to store the raw data files and facilitate data transfer.

3.3. Generating Pipelines

- I designed and implemented data pipelines in Azure Data Factory. These pipelines were responsible for extracting data from the source, transforming it as necessary, and loading it into the SQL Server.

4. Data Transfer to SQL Server

4.1. Setting Up SQL Server

- I created an Azure SQL Server instance to host the transformed data. I configured the database and tables based on the data schema of my shopping behavior dataset.

4.2. Data Movement

- I used the data pipeline in Azure Data Factory to move data from the storage account to the SQL Server. This involved setting up linked services and datasets within the data factory for both the source (storage account) and destination (SQL Server).

5. SQL Analysis

5.1. Querying Data

- I executed various SQL queries to analyze the data stored in the SQL Server. I focused on metrics like total sales, sales by category, and customer demographics.

5.2. Data Transformation

- I created views and transformed data using SQL, preparing it for more advanced analysis in Power BI. This included aggregating sales and segmenting customers based on their purchasing behavior.

6. Power BI Visualization

6.1. Loading Data into Power BI

- I connected Power BI to my SQL Server to load the transformed dataset into the reporting environment.

6.2. Creating Reports and Dashboards

- I created various visualizations, including:
 - **Total Sales:** Displayed using a Card visual.
 - **Sales by Category:** Represented using a Bar Chart.
 - **Sales Over Time:** Illustrated with a Line Chart.
 - **Customer Demographics:** Analyzed with Pie Charts and Bar Charts.

6.3. Advanced Analytics

- I performed clustering analysis using K-Means clustering to segment customers based on their purchasing behavior. I used Python within Power BI to execute the clustering algorithm and visualize the results using scatter plots.

7. Conclusion

This project has equipped me with valuable experience in handling ETL processes, data analysis, and visualization using Azure Data Factory, SQL Server, and Power BI. Through this comprehensive approach, I was able to transform raw data into actionable insights, ultimately improving my understanding of customer behavior and sales performance.