ETL (Extract, Transform, Load) Pipeline Documentation

December 2, 2023

Overview

This documentation outlines the ETL pipeline designed to handle data sourced from the Federal Reserve Economic Data (FRED). The dataset focuses on outstanding commercial real estate loans held by all commercial banks, measured in billions of dollars across various dates. The pipeline is implemented in Python using a Google Colab notebook. Afterwards, the transformed data will be loaded into Google BigQuery for further analysis. Since both are Google products, the entire pipeline creation process is more efficient.

Prerequisites

Before running the ETL pipeline on Google Colab, please do the following:

- Google Account: You need a Google account to access Google Colab and BigQuery.
- FRED API Key: Sign up for a FRED API account to obtain the API key. After signing up, obtain your API key from your FRED account.
- Google Cloud: Create a Google Cloud account and project. Then, enable the BigQuery API and obtain your project id from your Google Cloud account.

Note: Please make sure your Google Colab and Google Cloud are associated with the same email address.

ETL Process

Data Ingestion

Customizable Parameters

When interacting with the FRED API, you have the flexibility to customize the extraction process by changing various parameters. The following parameters are available for customization:

- `realtime_start`: Specifies the start date for real-time data retrieval.
- `realtime_end`: Specifies the end date for real-time data retrieval.
- `limit`: Sets the maximum number of data entries to extract.
- `sort_order`: Determines the sorting order of the data ('asc' for ascending, 'desc' for descending).

If you choose not to customize these parameters, default values are automatically applied from the default parameter dictionary during the data extraction process. After customization, data is extracted from the FRED API in JSON format.

Data Transformation

The transformation process is applied into two distinct data frames. The first data frame contains raw data such as date and loan amount. In the second data frame, the loan amounts are aggregated by year, and it consists of some fundamental insights such as the total, average, highest, and lowest values.

First Data Frame:

- Remove Redundant Columns: Eliminate redundant 'observation_start' and 'observation_end' columns, as these parameters were specified before data extraction.
- Column Renaming: Rename the 'value' column to 'loan_amount'.
- Data Type Adjustment: Change the data type of the 'loan_amount' column to `float64`.

Second Data Frame:

- Date Formatting: Convert the 'date' column to a datetime format.
- Adding Year Column: Introduce a new column, 'Year'and extract the year information from the 'date' column.
- Aggregation by Year: Aggregate the data by year, calculating various metrics such as sum, mean, max, min, and standard deviation for the 'Loan_Amount_in_billion' column.

Now, both data frames are ready. The next step is to load them into Google BigQuery for further analysis.

Data Loading

The two data frames are loaded to the target database, Google Big Query. To maintain consistency, the schema design for the BigQuery tables align with the data types of the existing data frames. The for loop will automatically determine the column types of the existing data frames and apply them to the new tables. To ensure the data loaded successfully, a query is then executed to print a few lines from the loaded data. The loading process is facilitated by the `to_gbq` function, specifying the project ID, dataset ID, table name, and an optional 'if_exists' parameter. The 'if_exists' parameter is helpful to handle cases where the table already exists. For more information about `to_gbq`, please see reference.

Running the ETL Pipeline on Google Colab

To execute the ETL pipeline successfully, download the provided Python script from GitHub and run each cell in sequence. This step-by-step process will make the pipeline run smoothly. After completing the pipeline, you can proceed to analyze the transformed data in Google BigQuery. Additionally, you have the option to export the data to platforms such as Looker Studio for advanced visualization and analysis.

Note: Prior to running the ETL pipeline, please have all the necessary prerequisites.

Troubleshooting

Please follow these steps:

- Make sure that you have installed the required libraries by executing the following commands:

```
!pip install pandas
!pip install google-cloud-bigquery
```

- Make sure to enter your Google Cloud project ID in the code.
- Make sure that your Google Colab email address matches the one you used to sign up for Google Cloud.

References/ Resources

- FRED API Series Observations Documentation
- Google Cloud BigQuery Python Client Library
- Documentation for to_gbq Function