ETL Report Guide

Pemba Sherpa, Raymond Sepulveda, Scott Partacz, Lindsey Oh
02/07/22

Introduction

Set the stage. Introduce the problem that you are trying to solve. Identify sources of data. Describe why the data needs to be transformed.

For this project our team chose to focus on the health insurance industry. Using U.S. Census data, we will categorize people into clusters based on disability and health status. With these clusters as the metric, we will then look at the relationship between socioeconomic factors and out-of-pocket costs for the insured. The socioeconomic factors of interest are age, race, income, and sex. This analysis is intended to provide the insured population in the U.S. with insights into how their out-of-pocket costs compare to those in a similar demographic. The U.S. Census data needs to be transformed because there are over 800 columns in our data sources, a majority of which will be unnecessary for this project.

Data Sources

Where did you find your data? When did you access it? Cite your sources here. Use proper citation here, using the APA format.

U.S. Census Bureau. (2021, October 8). *Annual Social and Economic Supplements*. Census.gov. Retrieved February 7, 2022, from https://www.census.gov/data/datasets/time-series/demo/cps/cps-asec.2021.html

U.S. Census Bureau. (2021, October 8). *Sahie API*. Census.gov. Retrieved February 7, 2022, from https://www.census.gov/programs-surveys/sahie/data/api.html

Extraction

Where did you get the data from? How did you get the data? What format is the extracted data? What steps were taken to extract the data? Be sure to number steps when the order matters.

1. Create a new Azure Data Storage Container in the 'gen10datafund2111' Azure Data Lake
2. Upload the CSVs from the census data source
3. Create a new Azure Databricks and set up the configuration variables for the Azure Storage Account
    a. Ensure that the Storage Account is 'gen10datafund2111' and the Storage Container matches the storage container that was named previously
4. Connect to the Azure Storage container using the previously defined configuration variables and display the file path
5. Extract the CSV from the Azure Data Storage container using the displayed file path and convert it into a Py Spark data frame

The data was downloaded as a CSV from the first data source. The CSV was read into a Jupyter notebook as a pandas data frame. The API data was called from the second data source and read into a Jupyter notebook as a pandas data frame.

Transformation

Did you use all of the data you extracted as-is? Did you remove columns? Did you change columns' names? Did you change your column formats? What steps were taken to get the data in a form that you could use it? Be sure to number steps when the order matters.

Since your end goal will be to load your data in SQL Server, include table mappings that identify the source data and its destination.

1. Drop all columns except:
    a. PERIDNUM (22-digit Unique Person identifier)
    b. PEHSPNON (Are you Spanish, Hispanic, or Latino?)
    c. PH_SEQ (Household seq number)
    d. AGE1 (Age recode - Persons 15+ years)
    e. PEAFEVER (Did you ever serve on active duty in the U.S. Armed Forces?)
    f. A_HGA (Educational attainment)
    g. A_SEX (Sex)
    h. PEDISDRS (Does this person have difficulty dressing or bathing?)
    i. PEDISEAR (Is this person deaf or does this person have serious difficulty hearing?)
    j. PEDISEYE (Is this person blind or does this person have serious difficulty seeing even when wearing glasses?)
    k. PEDISOUT (Because of a physical, mental, or emotional condition, does this person have difficulty doing errands along such as visiting a doctor's office or shopping?)
    l. PEDISPHY (Does this person have serious difficulty walking or climbing stairs?)
    m. PEDISREM (Because of a physical, mental, or emotional condition, does this person have serious difficulty concentrating, remembering, or making decisions?)
    n. PRDISFLG (Does this person have any of these disability conditions?)
    o. PRDTRACE (Race)
    p. A_MJIND (Major industry code)
    q. A_GRSWK (How much does this person usually earn per week at this job before deductions, subject to top coding, the higher of either the amount of item 25a times Item 25c or the actual item 25d entry will be present.)
    r. A_HRSPAY (How much does this person earn per hour?)
    s. A_CLSWKR (Class of worker)
    t. A_WKSTAT (Full/part-time status)
    u. PEARNVAL (total person's earnings)
    v. PTOTVAL (total person's income)
    w. AGI (Federal adjusted gross income)
    x. PERLIS (poverty level of person)
    y. COV (Any health insurance coverage last year)
    z. COV_CYR (Any coverage last year: none, some, all)
    aa. PUB (Public coverage last year)

bb. PUB_CYR (Public coverage last year: none, some, all)
cc. DEPPRIV (Private coverage through household member last year)
dd. GRP (Any employment-based coverage last year)
ee. DIR (Any direct-purchase coverage last year)
ff. MRK (Any Marketplace coverage last year)
gg. NONM (Any non-Marketplace coverage last year)
hh. CAID (Medicaid, PCHIP or other means-tested coverage last year)
ii. OTHMT (Other means-tested coverage last year)
jj. MCARE (Medicare coverage last year)
kk. MIL (Any TRICARE coverage last year)
ll. CHAMPVA (CHAMPVA coverage last year)
mm. VACARE (VACARE coverage last year)
nn. MOOP (Total medical out of pocket expenditures. Calculated from PHIP_VAL, POTC_VAL, and PMED_VA)
oo. MOOP2 (Total medical out of pocket expenditures. Calculated from PHIP_VAL2, POTC_VAL, and PMED_VA), PEMCPREM (Edited Medicare premium amount)
pp. PHIP_VAL (Out of pocket expenditures for comprehensive and non-comprehensive health insurance premiums)
qq. POTC_VAL (Out of pocket expenditures for over-the-counter health related spending)
rr. PMED_VAL (Out of pocket expenditures for non-premium medical care)
ss. PHIP_VAL2 (Out of pocket expenditures for comprehensive and non-comprehensive health insurance premiums - alternative (See https://www.census.gov/topics/health/health-insurance/guidance.html))
tt. POTC_VAL (Out of pocket expenditures for over-the-counter health related spending)
uu. PMED_VAL (Out of pocket expenditures for non-premium medical care)
vv. PECOULD (Eligible to purchase employer's health insurance plan)
ww. PEOFFER (Employer offers health insurance plan)
xx. HEA (Health status)

2. Filter rows for people who had any health insurance coverage the previous year.

Load

If you were to load your transformed data into a SQL database, what steps would you take to make that happen? Be sure to number steps when the order matters.

1. In Azure Data Studio, create a new SQL database on the gen10 server, create a new user and ensure that it has 'db_owner' permissions, so that you can access, query, and edit the database
2. Once the data is properly transformed, create a Producer Databricks that will take the transformed data from the Azure blob and send messages to Kafka utilizing these steps.
    a. Define and set up the Kafka variables needed for the OS and configuration
        i. Define the Confluent Topic Name as "algorhythm"
    b. Set up the admin client using previously defined Kafka variables
    c. Set up and define the producer using previously defined Kafka variables
    d. Produce a message to Kafka for every row in the transformed Data frame

        i. To simulate streamed data, a sleep command was implemented to send one message to Kafka every 5 seconds rather than all at once
3. Create a producer pipeline in Azure Data Factory to automate this process. Give the producer pipeline a storage event trigger to notify the team if there were any changes in the storage of the data in the Azure blob
4. Create Consumer Databricks that will consume the messages from Kafka and upload into our SQL database utilizing these steps
    a. Defined and set up the Kafka variables needed for the OS and configuration
        i. Ensure that the variables, especially the Confluent Topic Name, match the variables set in the Producer Databricks
    b. Set up and define the consumer using previously defined Kafka variables
    c. Create an empty list and empty dictionary for the consumer
    d. Consume each message from Kafka utilizing these steps
        i. Check to see if there are any messages to consume from Kafka
        ii. If there are messages in Kafka, consume the message and load it as a JSON into the empty dictionary
        iii. Create new element named 'Timestamp' was added to the dictionary, defined from the timestamp of the Kafka message
        iv. Append the dictionary to the list
    e. Convert the new list into a Py spark data frame and drop any empty/duplicate rows
    f. Define the connection strings for the SQL database:
        i. Database
        ii. User
        iii. Password
        iv. Server
        v. Table
    g. Insert the newly created data frame into the SQL database using 'write' and the previously defined SQL connection strings
5. Create a consumer pipeline in Azure Data Factory to automate this process. Give the consumer pipeline a schedule trigger that will trigger it to consume every 30 minutes (or other interval of time), or not run if there are no messages to consume in Kafka.

Conclusion

The data was successfully extracted from the data sources, transformed using pandas, and loaded into the SQL database. The loaded data will be used to create Machine Learning models and visualizations in Power BI.