**ETL Project**

Gilberto Ramirez, Ashok Singh, Chris McCown, Mohamed Badreddine

Version 1.0 | Aug 17, 2019

# **Project Summary**

The purpose of this project was to analyze the major factors which might affect the consumer financial complaints. To accomplish this, we first pulled the data from the following databases

* Consumer Complaint data from the Consumer Financial Protection Bureau Database
* Microeconomic Data from Federal Reserve Bank of New York
* Unemployment Data from the Bureau of Labor Statistics
* GDP per Capita data from Wikipedia database
* Two Letter State Abbreviations from USPS database

The dataset came in different formats, i.e. CSV file, excel file and HTML files. Some of the data sets were fairly large. For example, Consumer Complaint Database was 800MB database with 1.05M records. After assembling the raw dataset, we focused on transforming the dataset to the datasets of our interest. We focused on year 2018 because 2018 has the most consistent information across the different databases. Also, we focused on transforming the datasets by state. We used Python/Pandas libraries to transform the database. After transforming the datasets, we loaded the relevant dataset in Postgres SQL database using SQLalchemy connection.

# **Consumer Debt Database**

As part of this ETL project, we wanted to use public domain data defining the composition of consumer debt in the United States. Best source we could find for this data was the Center for Microeconomic Data from the Federal Reserve Bank of New York https://www.newyorkfed.org/medialibrary/interactives/householdcredit/data/xls/area\_report\_by\_year.xlsx

**Extract**

Since data that needed to be loaded was in a spreadsheet, we decided to use Pandas to fetch the contents (read\_excel function) from each sheet within the spreadsheet. During the extraction process, we had to skip first three rows. Also, the “Info” sheet had no relevant information we wanted to load in the database.

**Transform**:

There was consistency in all the table data. However, there was some redundant data we wanted to remove to make it more consistent with the rest of the tables in the database coming from other sources. One of the redundancies identified was in the column headers. Since all data is for end of fourth quarter of each year, we decided to remove the “Q4\_” string from all the column names using Pandas. A second and more involved transformation performed on the data is reshaping the data received. All sheets coming from the source have one column for the state, and the data is spread in several columns, one per year. However, to make it easier to manipulate once in database tables, we decided to consolidate each sheet in three columns: one for the year, one for the state, and onefor the values. That required data manipulation using Pandas DataFrame facilities like slicing, typecasting, and column renaming. Also, all the sheets were consolidated in just one data frame with this format, so we could have one column per data field to make easier calculations, and joins with other tables in rest of the database.

**Load**:

We used SQL Alchemy to create a connection to the database, and the to\_sql method from Pandas to do the uploading of the resulting DataFrame. Since we wanted to control the schema, atable was created defining couple of fields (year and state) as the primary keys, and the ‘append’ mode in the to\_sql method to avoid allowing Pandas to create a generic table based on defaults.

# **Consumer Complaint Database**

We got the Consumer Complaint Database from the Consumer Financial Protection Bureau. It had all the consumer complaint data from year 2014 to 2019 for all the states. It was a large data set with 1.05 records and 800MB.

**Extract**

Read the CSV file using Pandas read\_csv from <https://www.consumerfinance.gov/data-research/consumer-complaints> to a dataframe complain\_df.

**Transform**:

The database “date” was in a string format. Used Pandas “to\_datetime” and a lambda function to transform the “date” to “year”. Use Pandas to get the data for year 2018 and store in a df\_2018 dataframe. Used Pandas “value\_counts” function to count user complaint by state for df\_2018 dataframe.

**Load**:

Created an SQLAlchemy engine object and then wrote count\_df to

etl\_project database using Pandas to\_sql.

# **Unemployment Data from the Bureau of Labor Statistics**

**Extract**

Read the data using Pandas read\_html from https://www.bls.gov/news.release/laus.t01.htm to a dataframe unemployment\_df

**Transform**:

Used iloc to drop columns, only keeping the state and the 2018 unemployment rate. Used astype to convert ‘unemployment\_rate’ to float datatype.

**Load**:

Created an SQLAlchemy engine object and then wrote unemployment\_df to

etl\_project database using Pandas to\_sql routine.

# **GDP Per Capita and State Abbreviations Database**

**Extract**

Read the data using Pandas read\_html from <https://en.wikipedia.org/wiki/List_of_U.S._states_by_GDP_per_capita> and <https://en.wikipedia.org/wiki/List_of_U.S._state_abbreviations> databases

**Transform**:

Removed the unneeded columns. Changed the column names, transposing the data to get the needed data.

**Load**:

Created an SQLAlchemy engine object and then wrote to etl\_project database using Pandas to\_sql routine.

# **Project challenges and how we overcame**

* Getting the different datasets of our interest was a challenge. We had to look at multiple data sources, including data.world, Kaggle and other data sources. After scanning the multiple data sources, we finally settled on our datasets.
* Datasets came in different format, excel, CSV, HTM and for different years. Normalizing the dataset was a challenge. We normalized the data using Pandas dataframe and wrote the normalized data to Postgres SQL database.
* From HTML to Pandas, rather than getting one big table, Pandas was getting multiple tables. We combined all the tables into one.
* Postgres is case sensitive. While cleaning the data, we did not pay attention to the case-sensitivity and it was causing a problem during merging the tables. We had to later revisit our data cleaning and normalize for case sensitivity.

# Git Hub Project Location

* Github main project Repository

<https://github.com/cmccown1/ETL_Project>

# Data Sources

* Consumer Complaint Database from the Consumer Financial Protection Bureau

<https://www.consumerfinance.gov/data-research/consumer-complaints>

* Microeconomic National Data

<https://www.newyorkfed.org/microeconomics/hhdc.html>

* Unemployment Data from the Bureau of Labor Statistics

<https://www.bls.gov/news.release/laus.t01.htm>

* Wikipedia U.S. states sorted by gross domestic product (GDP) per capita

<https://en.wikipedia.org/wiki/List_of_U.S._states_by_GDP_per_capita>

* USPS Two Letter State Abbreviations

<https://pe.usps.com/text/pub28/28apb.htm>