Capstone ETL Report

Han Luong, Jed Dryer, Phil Carbino

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 our project, we are attempting to predict the affordability of housing for each county in New Jersey based on data related to median household income, median home prices, mortgage interest rates, and amount of new construction. Our data comes from three main sources:

- Zillow (for home price data)
- Freddie Mac (for mortgage rate data)
- US Census (for income and construction data)

Our goals for transforming the data sets were to distill the information down from a national level to a state and then county level for New Jersey. Once extracted, we had to transform the data to select only relevant columns, impute null values, standardize value types and formats, create separate bridge tables and merge tables where appropriate.

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.

- New construction building permit data from the U.S. Census:
  U.S. Census Economic Surveys Response. Index of /econ/BPS/county. (n.d.). Retrieved May 6, 2022, from https://www2.census.gov/econ/bps/County/
- Median house prices from Zillow:
  Housing Data. Zillow Research. (2021, March 25). Retrieved May 6, 2022, from https://www.zillow.com/research/data/
- Mortgage rates from Freddie Mac:
  30 Year Fixed Rate Mortgages since 1971. Retrieved May 11, 2022 from https://www.freddiemac.com/pmms/pmms30
- Median income data from US Census:
  Bureau, U. S. C. (2021, November 23).
  American Community survey 1-year data (2005-2020). Census.gov. Retrieved May 6, 2022, from https://www.census.gov/data/developers/data-sets/acs-1year.html.https://api.census.gov/data/<year>/acs/acs1/variables.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.

Building Permits:
Building permit data was scraped from the US Census website listed above. On this page, there are a series of links to .txt files, each of which was iterated through and scraped for data which was then transformed in a databrick and stored into an Azure blob.

Income:
Median income data was acquired via a US Census API call. For each year present, an API call was made to get a list of all the variables for the table that year. These variables were then converted into strings which were then used as end points for another round of API calls to get the data associated with the variables, which was then turned into a dataframe and transformed in a databrick as outlined below.

Mortgage Rates:
Mortgage rate data was scraped from tables directly in the body of the above website using BeautifulSoup. The data from each table was extracted from the created “soup” and then combined into dataframes and transformed in a databrick as outlined below.

Zillow Data:
Zillow home price data was extracted from a csv file obtained from the URL above. CSV was converted into a dataframe and transformed in a databrick as outlined below.

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.

Building Permits:
In a databrick, using python:

1. Drop unneeded columns
2. Renamed columns:
    a. ‘1-unitunits': '1_Unit',
    b. ‘2-unitsunits': '2_Unit',
    c. '3-4 unitsunits': '3-4_Units',
    d. '5+ unitsunits': '5_plus_Units',
    e. '_c12bldgs': '3-4_UnitBuilding',
    f. '_c15bldgs': '5_plus_UnitBuilding',
    g. '_c6bldgs': '1_UnitBuilding',
    h. '_c9bldgs': '2_UnitBuilding',
    i. 'countyname': 'County',

   j. 'fips1state': 'StateFips',
   k. 'fips2county': 'CountyFips',
   l. 'surveydate': 'Date'

3. Created “new_Date” column From “Date” column, changing format from "%y%M%d" to "yyyyMM".
4. Split “new_Date” column into “Month” and “Year” Columns
5. Convert ”Month” into a string value
6. Dropped old “Date” column
7. Renamed newly formatted date column
8. Combined “StateFips” and “CountyFips” into one merged “FIPS” column
9. Filtered data to only include records from New Jersey using “.like(34%)” on FIPS column
10. Replaced all null values with “0”
11. Added “NewUnits” and “NewBuildings” for overall numbers of units and buildings.
12. Dropped unnecessary columns
13. Left with:
    a. 'FIPS',
    b. 'County',
    c. 'Year',
    d. 'Month',
    e. 'NewUnits',
    f. 'NewBuildings',
    g. 'Date'
14. Save to blob as JSON file

Income:

In a databrick, using python:

1. Rename columns:
    a. 'Householder 25 to 44 years_inflation_adjusted_in_data_year': '25-44',
    b. 'Householder 45 to 64 years_inflation_adjusted_in_data_year': '45-64',
    c. 'Householder 65 years and over_inflation_adjusted_in_data_year': '65-plus',
    d. 'Householder under 25 years_inflation_adjusted_in_data_year': 'under-25',
    e. 'NAME': 'County',
    f. 'Total_inflation_adjusted_in_data_year': 'overall',
    g. 'YEAR': 'Year',
    h. 'county': 'CountyFips',
    i. 'state': 'StateFips'
2. Convert data from wide to long form
3. Combine “StateFips” and “CountyFips” into one “FIPS” column
4. Filter to include only data from New Jersey using “.like(34%)” on FIPS column
5. Split the “County” column to get only the county name
6. Save to blob as JSON file

Mortgage Rates:

In a databrick, using python:

1. For every table found in scraped data, get header row, to make table header
2. Add “Month” column to the table header
3. Get all rows from the tables
4. Create dataframe from the rows of each table
5. Set table header as column names
6. Add dataframe to a list of dataframes
7. Combine all dataframes into one large dataframe
8. Convert dataframe from wide to long form
9. Create a new column for the year
10. Create a new column to hold the attributes “rate” and “points”
11. Drop “YearAttribute” column
12. Rename the month column “PrimaryMonth”
13. Replace all blank values in “MortgageRateValue” with “None”
14. Split dataframe into two:
    a. One with “rate”
    b. One with “points”
15. Rejoin dataframes so that “rates” and “points” columns are now next to one another
16. Filter out “ANNUAL AVERAGE” values from “Month” column
17. Save to blob as JSON file

Zillow:

In a databrick using, using python:

1. Renamed columns:
    a. “RegionName” as “County”
    b. “MunicipalCodeFIPS” as “CountyFIPS”
2. Dropped columns:
    a. “RegionID”,
    b. “SizeRank”,
    c. “RegionType”,
    d. “Metro”,
    e. “StateName”
3. Convert from wide to long form
4. Split date into “Month” and “Year” columns
5. Combine “StateCodeFIPS” and “CountyFIPS” values into one “FIPS” column
6. Filter to include only data from New Jersey using “.like(34%)” on FIPS column
7. Save to blob as JSON file

Merging Tables:

1. Create a read mount point to blob where cleaned data JSON files are
2. Read in all cleaned data files from blob
3. Rename columns intended to be dropped from Zillow house prices data after table join:
    a. 'County_DROP',
    b. 'FIPS_Drop',
    c. 'Month_drop',

    d. 'Year_drop'

4. Rename columns intended to be dropped from mortgage data after table join:
    a. 'Month_Drop',
    b. 'Year_Drop'
5. Standardize Months across tables to be varchar(3) abbreviation format
6. Join building permits data and and house prices data on the following columns to begin creating main table:
    a. building_permits.FIPS == house_prices.FIPS_Drop
    b. building_permits.Year == house_prices.Year_drop
    c. building_permits.Month == house_prices.Month_drop
7. Join newly created main table (above) with mortgage data on the following columns:
    a. main_table.Month == mortgage_data.Month_Drop
    b. main_table.Year == mortgage_data.Year_Drop
8. Drop all columns that have been renamed to express “drop” in their name
9. Break “Year” column out into its own table
10. Break “Month” column out into its own table
11. Break “County” column out into its own table

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.

To load the data into a SQL database, we first created the following schema:

1. main_table
    a. FIPS (PK, FK, varchar(7), not null)
    b. YearID (PK, FK, int, not null)
    c. MonthID (PK, FK, int, not null)
    d. NewUnits (int, not null)
    e. NewBuildings (int, not null)
    f. MedianHousePrice (int, null)
    g. AverageRate (numeric(5,2), not null)
    h. AveragePoints (numeric(4,2), not null)
2. county
    a. FIPS (PK, varchar(7), not null)
    b. County (varchar(100), not null)
3. median _income
    a. FIPS (PK, FK, int, not null)
    b. YearID (PK, FK, int, not null)
    c. AgeGroup (PK, varchar(20), not null)
    d. MedianIncome (int, not null)
4. month
    a. MonthID (PK, int, not null)
    b. Month (varchar(3), not null)
5. year
    a. YearID (PK, int, not null)

b. Year (int, not null)

We then had the databrick which previously merged the dataframes into their final form write the data for the “county”, “median_income”, “month”, and “year” tables to the target tables in the SQL database. The “main_table” data was used by our kafka producer and consumer, and then sent to the target table in the database.

Conclusion

Wrap it up here.

At this point, primary and foreign keys have been established, cleaning was accomplished during the transformation process, and we now have our databases fully loaded and ready to use for our, ML models and visualizations.