Repeatable ETL Report

Ahad Hussain, Sharif Rakhimov, Luis Rivera, Marjea Mckoy, Parth Patel

Introduction

Our group is looking to predict the likelihood of someone being uninsured based on where they are and their demographic category. We use data sources from the Census Bureau and the United States Department of Agriculture. The datasets contain several columns which we do not need, which the report discusses below.

Extraction

All the data sources can be found in our "Data Sources.pdf" in the Project Specification folder. All the datasets were uploaded into our storage container in gen10datafund2202 in the pushing-p-breakers container.

Data Sources

References

US Census Bureau. (2021b, October 8). *Cartographic Boundary Files* - Shapefile. Census.Gov. https://www.census.gov/geographies/mapping-files/time-series/geo/carto-boundary-file.html

U.S. Census Bureau. (2020d). *Explore Census Data. Census.Gov.* https://data.census.gov/cedsci/table?q=new%20jersey%20uninsured%20by%20census%20tract&g=0400000US34%241400000

City-Data.com - *Stats about all US cities* - real estate, relocation info, crime, house prices, cost of living, races, home value estimator, recent sales, income, photos, schools, maps, weather, neighborhoods, and more. (2020). City-Data. http://www.city-data.com

U.S. Census Bureau. (2020a). *Explore Census Data*. Income. https://data.census.gov/cedsci/table?q=county%20subdivision%20new%20jersey%20income&tid=ACSST5Y2020.S1901

U.S. Census Bureau. (2020b). *Explore Census Data*. Unemployment. https://data.census.gov/cedsci/table?q=county%20subdivision%20new%20jersey%20unemployed&tid=ACSDP5Y2020.DP03

U.S. Census Bureau. (2020c). *Explore Census Data*. Uninsured. https://data.census.gov/cedsci/table?q=county%20subdivision%20new%20jersey%20uninsured&tid=ACSST5Y2020.S2701

US Census Bureau. (2021, October 8). *2008 - 2019 Small Area Health Insurance Estimates (SAHIE) using the American Community Survey (ACS)*. Census.Gov. https://www.census.gov/data/datasets/time-series/demo/sahie/estimates-acs.html

U.S. Census Bureau. (2020e). *Explore Census Data*. US Census Bureau.

https://data.census.gov/cedsci/table?q=unemployment&g=0400000US34%241400000&tid=ACSDP5Y2020.DP03

U.S. Census Bureau. (2020f). *Explore Census Data*. US Census Bureau.

https://data.census.gov/cedsci/table?q=unemployment&g=0400000US34%240500000&tid=ACSDP5Y2020.DP03

Transformation

First, we must establish a mount point to a storage container. Later, this will be used to write the final table to the storage container. Once the mount point has been established, the NJ HealthCare Data, NJ Income by City, NJ Unemployment, NJ Uninsured, and SAHIE 2019 tables will be there. The transformation is below for each table and merging them:

Uninsured Cities

1. Read the NJ_Uninsured CSV file into a pandas dataframe
2. Create a list of columns to be selected: NAME, S2701_C01_001E, S2701_C04_001E, S2701_C04_002E, S2701_C04_003E, S2701_C04_004E, S2701_C04_005E, S2701_C04_006E, S2701_C04_007E, S2701_C04_008E, S2701_C04_013E, S2701_C04_014E, S2701_C04_015E, S2701_C04_016E, S2701_C04_017E, S2701_C04_018E, S2701_C04_019E, S2701_C04_020E, S2701_C04_021E
3. Drop the first row from the dataframe
4. Select the listed columns interested
5. Remove rows where the column NAME contains County subdivisions not defined
6. Split the Name column into three columns by the comma delimiter. The zero index is a city, the first index represents the County column, and the second index represents the State column
7. Remove the repeated word city in the city column
8. Convert the City column format to a title format and drop the original City column

9. Rearrange the columns so the first three columns are State, County, and City, with the rest being in the same position
10. Rename the columns: S2701_C01_001E: City Population, S2701_C04_001E: Uninsured Population, S2701_C04_002E: NUI Under 6 years, S2701_C04_003E: NUI 6-18 years, S2701_C04_004E: NUI 19 to 25 years, S2701_C04_005E: NUI 26 to 34 years, S2701_C04_006E: NUI 35 to 44 years, S2701_C04_007E: NUI 45 to 54 years, S2701_C04_008E: NUI 55 to 64 years, S2701_C04_013E: NUI 65 years and older, S2701_C04_014E: NUI Men, S2701_C04_015E: NUI Women, S2701_C04_016E: NUI White, S2701_C04_017E: NUI African American, S2701_C04_018E: NUI American Indian & Alaska Native Population, S2701_C04_019E: NUI Asians, S2701_C04_020E: NUI Native Hawaiians & Pacific Islanders Population, S2701_C04_021E: NUI Other Races
11. Convert the Uninsured Pandas dataframe to a Spark dataframe
12. Cast all the Numeric columns to an Integer Type
13. Convert the State, County, and City columns to a String Type

Income Cities

1. Read the Income CSV file into a Spark dataframe
2. Select the Name and S1901_C01_012E columns from this table
3. In the Name column, remove where the row contains 'County subdivisions not defined.
4. In the S1901_C01_012E, replace where the row contains '250,000+' to '250_000'.
5. Four median household income values are missing; Replace the values using the City-Data website:
    a. Teterboro borough, Bergen County, New Jersey: 39_196
    b. Tavistock borough, Camden County, New Jersey: 89_990
    c. Seaside Heights borough, Sussex County, New Jersey: 61_256
    d. Walpack Township, Sussex County, New Jersey: 88_407
6. Convert the columns S1901_C01_012E to an Integer Type
7. Split the Name column into three columns by the comma delimiter, with the zero index being City. The first index represents the County column, and the second index represents the State column.
8. Remove the repeated word city in the city column.
9. Convert the City column to a Title format and drop the original City column
10. Rename the initial(City) to City and S1901_C01_012E to Median Household Income
11. Rearrange the columns in the order of State, County, City, and Median Household Income
12. Convert the State, County, and City columns to a String Type

Unemployment Counties

1. Read the NJ_Unemployment_by_County CSV file into a spark data frame
2. Select the NAME, DP03_0062E and DP03_009PE columns

3. Drop the first row
4. Remove the rows that contain County subdivisions not defined in the NAME column.
5. Cast the NAME column to a String Type, DP03_0062E and DP03_009PE to a Float Type
6. Split the NAME column into two columns by the comma delimiter. The zero index is the County and the first index is the State
7. Rename the columns from DP03_0009PE to Unemployment Rate (16 & Over) and DP03_0062E to Median Household Income
8. Rearrange the columns in the order of State, County, Median Household Income, and Unemployment Rate (16 & Over)

Unemployment Cities

1. Read the NJ_Unemployment CSV into a spark data frame
2. Select the NAME, S2301_C04_001E columns
3. Drop the first row
4. Remove the rows that contain County subdivisions not defined in the NAME column.
5. Cast the NAME column to a String Type and the S2301_C04_001E to a Float Type
6. Split the NAME column into three columns by the comma delimiter. The zero index is City, the first index represents the County column, and the second index represents the State column
7. Remove the repeated word city in the city column.
8. Convert the City column to a Title format and drop the original City column
9. Rename the initcap(City) to City and S2301_C04_001E to Unemployment Rate (16 & Over)
10. Rearrange the columns in the order of State, County, City, and Unemployment Rate (16 & Over)

Unemployment Tract

1. Read the NJ_Unemployment_by_Cencus_Tract CSV file into a spark data frame
2. Select the NAME, DP03_0009PE, and DP03-0062E columns
3. Remove rows where the column NAME contains County subdivisions not defined.
4. Change the numerical columns to a Float Type
5. Split the NAME column into three columns by the comma delimiter. The zero index is Census Tract, the first index represents the County column, and the second index represents the State column
6. Rename the Columns from DP03_0009PE to Unemployment Rate (16 & Over) and DP03_0062E to Median Household Income
7. Rearrange the columns in the order of State, County, Census Tract, Unemployment Rate (16 & Over), and Median Household Income

SAHIE

1. Read the SAHIE CSV into a pandas dataframe and skip the first 79 rows

2. Convert the dataframe from pandas to spark
3. Right trim the empty spaces from state_name and county_name
4. Filter the agecat, sexcat, racecat, iprcat all to equal 0
5. Filter the geocat to equal 50
6. Group the data frame by NUI and cast the NUI to an Integer Type
7. Select the state_name, county_name, and NUI columns
8. Convert the data frame to a pandas data frame
9. Convert the state names to their abbreviations
10. Convert from a pandas data frame to a spark data frame
11. Rename the columns: state_name to State, county_name to County, and NUI to Number of Uninsured (2019)

After cleaning the datasets and transforming them, the tables are now ready to be merged.

Merge 1

1. Merge the Unemployment Cities and Income Cities data frames with an inner join on the State, County, and City columns
2. Merge the Unemployment and Income data frame with the Uninsured Cities dataframe with an inner join on the State, County, and City columns

Spatial Area Interpolation (Machine Learning Model)

1. Read the cb_2018_34_cousub_500k file into a variable using geo pandas.
2. Select the NAME, geometry, and COUSUBNS columns
3. Read the NJ_FIPS_codes CSV file into a variable.
4. Drop the rows where 5 contains County subdivisions not defined
5. Select columns 3, 4, and 5 from the table
6. Rename the NJ_FIPS_codes columns from 3 to County, 4 to COUSUBFP, and 5 to City
7. Convert the COUSUBFP column to a String Type
8. Pad the COUSUBFP column to 5 place holders
9. Remove the repeated word city in the city column.
10. Convert the City column to a title format
11. Merge the NJ_FIPS_codes and the geo pandas table with an inner join on COUSUBFP
12. Drop the NAME and COUSUBFP columns
13. Add a State column that contains New Jersey in every cell.
14. Order the columns in the order of State, County, City, and geometry
15. Convert the geo data table to a pandas dataframe
16. Sort the master table alphabetically by City
17. Create a new column that takes the geometry column and adds the column to the geo data table
18. Convert the table to a geo data frame
19. Read the census tract file (cb_2018_34_tract_500k) in a variable
20. Read the NJ_Uninsured_by_Census_Tract CSV file in a variable.

21. Drop the first row of the NJ_Uninsured_by_Census_Tract table
22. Rename the columns: NAME:Census Tract, GEO_ID: AFFGEOID, S2701_C01_001E: City Population, S2701_C04_001E: Uninsured Population, S2701_C04_002E: NUI, Under 6 years, S2701_C04_003E: NUI, 6-18 years, S2701_C04_004E: NUI, 19 to 25 years, S2701_C04_005E: NUI, 26 to 34 years, S2701_C04_006E: NUI, 35 to 44 years, S2701_C04_007E: NUI, 45 to 54 years, S2701_C04_008E: NUI, 55 to 64 years, S2701_C04_013E: NUI, 65 years and older, S2701_C04_014E: NUI, Men, S2701_C04_015E: NUI, Women, S2701_C04_016E: NUI, White, S2701_C04_017E: NUI, African American, S2701_C04_018E: NUI, American Indian & Alaska Native Population, S2701_C04_019E: NUI, Asians, S2701_C04_020E: NUI, Native Hawaiians & Pacific Islanders Population, S2701_C04_021E: NUI, Other Races
23. Split the Census Tract Column into 3 Columns by the comma delimiter into Census Tract, County, and State columns
24. Merge the Census tract file and NJ_Uninsured_by_Census_Tract with a right join on State, County, and Census Tract
25. Merge the Census tract file and NJ_Uninsured_by_Census_Tract with a right join on AFFGEOID
26. Convert the merged table to a geo data frame
27. Cast all the numeric columns to an Integer Type
28. Drop columns AFFGEOID, GEOID, NAME, LSAD, ALAND, AWATER, STATEFP, COUNTYFP, TRACTCE

Load

The data frame will be loaded into an SQL database. The steps on how to load this table into the database are below:

1. Ignore user warnings that contain geometry columns
2. Convert plygonal geometries to well-known binary hex string
3. Convert all the tables into a spark data frame

Producer

1. Define callback errors and raise exceptions if errors occur
2. Delete the topic if it already exists
3. Create a confluent topic to send messages to.
4. Establish your configuration for the SQL database
5. Create three separate tables for NJ Cities, NJ Census Tracts, and NJ County Info
6. Write each table into the SQL Server Database
7. Send each row of the to the producer for the NJ Cities and NJ Census Tracts

Consumer

1. Define callback errors and raise exceptions if errors occur
2. Make connection strings with the same confluent topic as the producer confluent topic
3. Create a Kafka consumer class setup

4. Consume messages from the topic with a timestamp
5. Append the messages in a dictionary
6. Create a spark dataframe from the dictionary with all the messages
7. Configure to which SQL Server Database you will be sending the data frame
8. Write to the SQL Database

Conclusion

All the data sources we used will be merged into a final data frame which will then be sent into an SQL database. Our group uses this table to predict the number of uninsured people within a geographic area. All the columns will be used in our SQL server to draw our conclusions and run a machine learning model.