W205 Final Project Progress Report
November 15th, 2016

`where(ii)`

By Melanie Costello, Lisa Barcelo, and Priya Gupta

Accomplishments

- ❏ Finalized Data sources: Zillow, Census, Texas A&M University Dept of Transportation
- ❏ Created ERD
- ❏ Created DDL scripts to clean/load data into Hadoop
- ❏ Updated ERD to connect tables and create structure of new tables
- ❏ Finalized ETL to transform data into appropriate tables (partitioning)

Highlights

Since we are aggregating household rental data for different types of properties (1 bed, studio, etc) spanning several decades, we decided to partition our rental pricing table by property type. Since our raw data was in separate tables, we were able to transform the data into sub tables and create an empty table with a pre-partitioned **bed_df** column. This allows us to quickly query the rental pricing table.

Because we are uniting multiple data sources, we needed to create a mapping of the identifiers in each source. Two of our primary identifiers are **metro_ID** from the Zillow data and **CBSA codes** from our income data. In cleaning the census data, which is the source of our income information, we noticed the mapping of CBSA to Zillow **metro_ID** would not be congruent. We experimented with various edit distance functions, and ultimately wrote a python script to appropriately align various regions and CBSA codes with the appropriate Zillow **metro_ID**.

Obstacles

We ran into some problems trying to make an edit distance function work for our mapping of CBSA to Zillow **metro_ID**. The main problem was that the names we were trying to align were of very different lengths (e.g., “Dallas, TX” vs. “Dallas Metropolitan Area”). Adjusting the scoring threshold to get these types of names to line up resulted in too many false positives. Instead, we ended up writing code to parse and compare the name fields.

A related issue centered around the lack of a common key. Not every data set had a “**region_id**” field. In order to resolve this, we used the Zillow data as our guide. We wrote a fuzzy string matching function to pair raw **Region Name** data with

corresponding regions in our guide data, and for any regions that existing in both our raw data and our guide data, we included the **region_id** in an adjacent dataframe column.

Outstanding

- ❏ Aggregating data into final display (formulas, time series, etc)
- ❏ Connecting Hadoop and Tableau
- ❏ Creating Tableau dashboards (open up server to make viewable to others)
- ❏ Use case testing

When planning this project, we initially decided to build a AWS web application on top of an EC2 instance. However, as a result of being introduced to Tableau, we decided to create an interactive dashboard that will enhance the user experience as well as provide both design flexibility and computing power during development.

In addition to making the Tableau connections, we need to further test various other kinds of queries. Finally, we will need to update our ERD to reflect the structural changes made after deciding to partition the table containing rental data.