Capstone Data Wrangling Project

Introduction

This report describes the steps I took to clean and prepare my data for use in my capstone project. The goal of the project is to predict AirBnB listing price and utilization rate using data taken from Inside AirBnB. This data will be merged with business data taken from Yelp's Academic Business dataset to determine the number of businesses and their popularity within a certain range of each AirBnB listing. The Capstone Proposal available here provides more information.

Note: The Jupyter Notebook containing all code is available here. The raw and cleaned data files are available here.

- Data_files.zip: zip containing the original AirBnB and Yelp files, in .csv and .json format respectively. The AirBnB dataset is 66MB, the Yelp dataset is 142MB.
- Listings_clean.csv: Cleaned version of the AirBnB dataset.
- Yelp_clean.csv: Cleaned version of the Yelp dataset.

Cleaning the AirBnB Listings Dataset

Initial Inspection and Removing Redundant Fields

The initial dataset contained 17,542 records and 96 columns. My first step was to remove fields that do not seem useful for various reasons. For example I dropped all columns related to reviews. My model is meant to provide new hosts (people that list their property on AirBnB are referred to as hosts) with a tool for deciding the optimal price to set their listing at. Therefore basing the model on reviews would not help, as new listings do not have reviews yet. The exception being reviews_per_month. This field can be used to estimate how heavily utilized a listing is, which is going to be the second stage of my model's predictions.

I also checked for and removed columns where all records contained a single value. For example last_scraped, which is the date the data was scraped by Inside AirBnB, and is_business_travel_ready, which consisted entirely of the value "f". Altogether dropping unnecessary variables, and those with a single value, brought me from 96 to 42 variables.

Null Values

My next step was to look for null values. 15 columns had missing data, ranging from a few records (host_is_superhost) to one column that was 100% missing (neighbourhood_group_cleansed). I dropped the 100% missing column entirely. A few variables were between 86-98% missing, so I deleted these columns and replaced them with a binary flag variable which indicates if there was a missing value. The flag variable can be fed into a predictive model.

For the remaining variables with nulls, I either replaced missing values with 0 (for example security_deposit, where I assumed missing meant it was not charged), "Unknown" (for text fields like neighborhood), or the most common value or mode (for example bathrooms, and host_is_superhost). I wrote a function to handle these missing values programmatically. The function takes a dataframe, a list of columns, and arguments "how" (drop or impute), "imp_num_method" (defaults to mean, this determines the statistic to replace numeric variable nulls with), and "imp_with" (defaults to most_common, this determines what to replace nulls with in categorical variables).

Outliers

I used pandas .describe() method with the percentiles argument to view the mean, median, standard deviation, min/max, and the 1st, 10th, 50th, 90th, and 99th percentiles. I used these to spot outliers in the data. I do this by looking for wide gaps between the min and 1% (or 10%), the median and mean, or the max and 99% (or 90%). Wide gaps between these numbers, usually more than 2-3 standard deviations, are a good indicator that outliers exist.

I generally handled outliers by trimming them to the 99th or 90th percentile value (or 1st/10th for outliers at the low end of the scale). For a few variables like price, which had a relatively small number of records above $2000 (around 200 out of 17K in the dataset) I decided to drop all records above the 99th percentile. The mega-expensive properties were not indicative of the majority of listings and I felt they would confuse my model. I also created a function for handling outliers programmatically, with arguments to trim or drop depending on need.

With these changes completed, the final dataset contained 17,269 records and 51 variables. I saved this to a .csv file for easier access later.

Cleaning the Yelp Business Dataset

Initial Inspection and Removing Redundant Fields

This dataset required much less cleaning than the AirBnB data. This is a .json file so I opened it with pd.read_json. The initial dataset had 188,593 records and 15 columns. The raw file has over 100 cities in it, but 6 make up the vast majority of records. The city of Toronto, which I have decided to analyze with this project, makes up 18,233 records. I saved those records and deleted the rest.I also removed closed businesses (where is_open = 0), bringing the dataset to 14,023 records.

Next I dropped columns I do not anticipate using in my modeling. My goal with this file is to create a variable for each AirBnB listing that shows the number of businesses and average star rating within a certain radius, not yet defined but likely .1 or .25 mile. The Yelp dataset has Neighborhood and Lat/Long, both of which I can use to match with the listing data. Therefore I do not need fields like address, categories, city, name, etc. I was able to drop 9 variables, bringing the total in the dataset from 15 to 6.

Null Values

Checking for Nulls initially returned no missing values, but visually inspecting the data I noticed blanks in the neighborhood column. I used the .value_counts() method to determine this is actually the most common neighborhood, around 20% of records are missing neighborhood. I plan to use this variable to limit the number of lat/long matches required when merging with the AirBnB data (to improve performance). I don't want to delete these missing records though, as they represent too much data to lose. For now I replaced the "" with "Unknown". No other values had missing data.

Outliers

Using .describe() showed no outliers in the data, except for review_count. I decided not to change these because I will eventually group by location, and I want to see if an area has a high volume of reviews.

As a final step I output the cleaned dataset to a .csv file for easier access in the future.