Capstone Project 1: Data Wrangling

Github Jupyter Notebook Link:
https://github.com/dtse91/Springboard/blob/master/Capstone%201%20Project/Capstone%20Project%201%20Data%20Wrangling.ipynb

1. **What kind of cleaning steps did you perform?**

The data was downloaded as of May 9th, 2018 from NYC Open Data. The .csv data contains a list of permits issued for a particular day and associated data. Prior weekly and monthly reports are archived at DOB and are not available on NYC Open Data. The raw building permit data contains 60 unique columns and nearly 3.37 million rows of data. A separate .csv file provides additional detailed descriptions about the data.

The data was then imported into a Pandas DataFrame for ease of data manipulations. Feature names were adjusted to be short yet meaningful, free of spaces via replacement using underscores and converted to lowercase. Twenty features out of the original 60 were kept for use in the EDA and machine learning stages to keep only relevant features in the scope of the study. Feature relevance was based on visual inspection of the records and partially due to the proportion of null/empty records. The raw data contained building permits issued dating back to the 1980s, however I decided to only use data from the past five years because it is likely to be more relevant from a prediction standpoint; processes have likely changed to reduce building permit issue times. However, I may use permits issued up to 10 years ago if additional data is required to correct class imbalance issues at the machine learning stage.

https://data.cityofnewyork.us/Housing-Development/DOB-Permit-Issuance/ipu4-2q9a

2. **How did you deal with missing values, if any?**

Missing values were primarily due to whether the feature is required or optional for a building permit. Even if the data is required, sometimes null values were still present as a small proportion of all of the records. In Python, specifically Pandas, NumPy and Scikit-Learn, the standard practice is to mark missing values as NaN. Values with a NaN value are ignored from operations like sum, count, etc. We can mark values as NaN easily with the Pandas DataFrame by using the replace() function on a subset of the columns we are interested in. After we have marked the missing values, we can use the isnull() function to mark all of the NaN values in the dataset as True and get a count of the missing values for each column.

After marking these records as NaN, I decided that it was acceptable to remove records where the proportion of nulls to total records is extremely small, e.g. the nulls in the latitude feature were <1% of the total number of records. For other features, I’ve decided to use a decision tree classifier to automatically categorize the empty records into their own category and calculate their log likelihoods, which avoids data imputation, data removal, and throws no errors using machine learning algorithms. This decision tree approach will be done at the EDA stage.

3. **Were there outliers, and how did you handle them?**

Outliers may be due to measurement/input error, data corruption, or perhaps actually represent legitimate behavior. I used Tukey fences as a simple way to determine outliers, i.e. a record is considered an outlier if it is 1.5 times the interquartile range. The dependent variable, issue time (days), is heavily right skewed based on its histogram. From the histogram, I observed negative values for the issue time, which I subsequently removed from the analysis because of suspected input error. On the other hand, outliers are present for permits requiring multiple years to be issued, and I suspect these represent legitimate records. For example, new buildings, especially high-rise and skyscrapers, typically require a much longer time to obtain a building permit compared to an average project. From the data, I notice that in the last five years some of these projects needed up to four years to obtain a building permit. One approach is to simply trim the bottom 1-5% of data, but there may be value in understanding the factors that lead to multi-year building permit issue times. However, it would be a good idea to trim permits with less than a month to issue. Permits with issue times less than a month are primarily from “quick-fix” projects, e.g. minor electrical work, which are generally more predictable and likely don’t provide much value to the client if included in this study.