Data Wrangling on Building Permit Data

A Part of the Capstone Project Submissions

Introduction

A building permit is an official approval document issued by a governmental agency that allows you or your contractor to proceed with a construction or remodeling project on one's property. For more details click here. Each city or county has its own office related to buildings, that can do multiple functions like issuing permits, inspecting buildings to enforce safety measures, modifying rules to accommodate needs of the growing population etc. For the city of San Francisco, permit issuing is taken care by Permit Services wing of Department of Building Inspection (henceforth called DBI). The delays in permit issuance pose serious problems to construction industries and later on real estate agencies. Read this. Trulia study and Vancouver city article.

This Document

This document particularly describes the data wrangling steps that I undertook to clean the capstone project data set. It explains what are tricky parts of data wrangling phase in general. What kind of cleaning steps were performed on this particular set, how the missing values were handled and, if any and if there were outliers, if yes how were they handled

Tricky Part!

The Tricky part of data wrangling,

1. Knowing what is present in the 'null' cells, is it NaN or simply ' ' or whitespace(s)
2. In the non-null cells, if the all values are meaningful
3. Recognizing that even if a column has all non-null and meaningful values, the future updates to the column may have problems. Hence need to expect it and handle it
4. See the data and think if certain value make sense for the business and decide to drop those which are not relevant

Cleaning Steps Performed:

1. **Identifying the columns to retain:** It is not smart to waste time cleaning up columns that may not turn out useful in modeling. Hence I chose to eliminate some in the first pass. I would err on the side of caution, hence kept a few extra ones, which might disappear in the later stages.
2. **Interpreting too small values for costs:** Cost of the project is an essential part of the application according to this post. Hence if any of the cost related columns, in this case, Estimated cost and Revised Cost have 0 or unusually small numbers, the best option is to drop those rows. Those are the outliers.
3. **Converting Invalid weekdays:** DBI is open only from Monday-Friday. What to do we do if filed date or issue dates map to Saturday or Sunday? I attributed it to typing mistake and make it previous or next day respectively. This may not be accurate (as typo may not be from n to n+1 or n-1 in day part alone, however it will at least prevent outliers in the EDA part.
4. **Filling the blanks in variables having only 'Y' as the entries:** In the application forms (both physical or online), normally the applicant is supposed to tick the option

if applicable. Otherwise nothing needs to be done. Hence it is understandable that blanks mean not applicable, a "No". Filled Fire only permit, Site Permit and Structural notification blank entries with “N”

5. **Filling nulls in qualitative variables with > 2 categories, which has numbers as entries:** I chose to fill these with a number which is already non-existing in the variable: For example, filled NaN’s in existing and proposed construction types with 9’s. Existing and proposed number of stories with 200.
6. **Filling nulls in qualitative variables with > 2 categories, which has strings as entries:** Replaced such variables with a more explicit “Unknown” instead of general NaN.
7. **Converting Location (Longitude / Latitude) into numbers:** NaN’s in this variables are substituted with 0’s. They are the same rows for which supervisor district and neighborhoods are unknown. We might end up retaining only only one of these three in the next stages. Longitude and Latitude were read as string types, so converted them to float numpy array of 2 elements.
8. **Dropping meaningless rows:** Dropping rows without corresponding issue date for the file date is important because only if both are present, wait time is a timedelta number and otherwise it is of no interest. Note that this step will invariably give columns with all valid dates for file date and issue dates.
9. **Saving the Clean file:** After all the cleaning steps, I wrote the dataframe to csv file so that the next step can pick it from there if needed.

The code for data wrangling is here