Capstone Project Report

# Step 1: Clean up data

The coordinates in the given data were noisy. For example, a single building might be recorded with different coordinates, which made data processing and feature engineering challenging.

After visualizing and analyzing pair-wise distance between the coordinates, I realized that it was reasonable to identify unique buildings based on the distance between buildings. In particular, a pair of coordinates identified a unique building if there were no another pairs of coordinates within 2 meter from it. If there were other pairs of coordinate within 2 meters range, all these pairs of coordinates should belong to a single building and only a pair of them was kept for future processing.

In addition, many incorrect coordinates which were obvious outside Detroit were removed.

Python library geopy was used in this step.

# Step 2: Define positive and negative data

After cleanup in Step1, there were around 5,000 unique pairs of coordinates identified in demolition permit data. These constituted the positive data for training and testing.

On the other hand, ~5000 unique pairs of coordinates were randomly selected from blight violation data. These served as the negative data for training and testing.

Since geopy was not very scalable, I had to implement a very fast distance calculation between two pairs of coordinates, considering that all pairs of coordinates were within a small flat area compared to the curvature of the Earth.

# Step 3: Feature engineering

Features for classification were aggregation from a set of categorical parameters in blight violations, 311 calls, and crime data. Given a categorical parameter and a building, a Python script calculates the counts of all categories around the building within 20 meters, 200 meters, and 2000 meters respectively. The count of a category within a range was treated as a feature.

The following is the list of parameters used for feature extraction:

* In blight violation data:
  + ‘ViolationCode’
  + ‘PaymentStatus’
  + ‘ViolationCategory’
  + ‘JudgementAmt’ (categorized into <$200, between $200 and $500, and >$500)
* In 311 calls:
  + ‘issue\_type’
* In crime data:
  + ‘CATEGORY’

990 features were generated from the above step.

# Step 4: Training a Classifier

Both Random Forest classifier and Extra Trees classifier were experimented with 5-fold cross- validation. It turned out that Random Forest classifier outperformed Extra Trees in this situation. Therefore, I focused on Random Forest classifier in the following.

After tuning the super parameters of Random Forest classifier, the best classification score of 5-fold cross-validation was 0.70.

In order to evaluate the benefits of feature engineering, all features used for classification were sorted by its importance. Starting with the more important features and adding more feature sequentially, I compared the performance of a subset of top features. It turned out that the top 100 features could reach a comparable performance as the one with 990 features.

# Opportunities

The 5-fold cross-validation score from the previous step was 0.70, which was not great for a classification problem. Several opportunities have been identified as below.

1. Feature engineering was limited by availability of data in various aspects.
   1. There was lack of information about the buildings, e.g. type of building (condominium, single family, townhouse, …), age, lot size, square footage, number of bedrooms, and so on.
   2. A feature was the count of a given category of a parameter within a given distance. However, this handling didn’t take into account the density of building. Therefore, the value of a feature might be impacted by the density of buildings about the given building but not necessarily hold predictive power.
2. In the current implementation, all incidents were used for feature extraction regardless of their temporal dependency. A more reasonable implementation would only use the incidents that occurred before demolition for feature extraction.
3. In addition to imperfect coordinates, the parameters used for feature extraction were also noisy. For example, there were multiple categories related to ‘ViolationCode’ ’22-2-83’ in blight violation data, say ’22-2-83’, ’22-8-83(a)(b)(c)’, ’22-2-83(a)(c)’, ’22-2-83(b)’, ’22-2-83(c)’, ’22-2-83(d)’, and ’22-2-83a’. There sub-categories should have been integrated or further partitioned.