Section 1: Week 2: Evaluate Data Mining Techniques

Nate Bachmeier

TIM-8130: Data Mining

February 15th, 2020

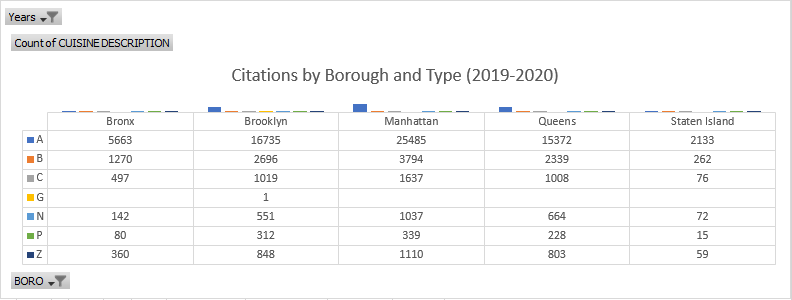
North Central University

Evaluate Data Mining Techniques

# Section I: Using Data Mining

The city of New York offers multiple data sets for businesses and researchers to explore and make discoveries. The Department of Health and Mental Hygiene publishes a daily feed of NYC Restaurant Inspection Results that enumerates all citations since July 2015 (DOHMH, 2020). Each establishment receives a grade of A, B, C, P(ending), N(ot available), or Z based on the severity of citations. According to NYC Health (2020), “a business must maintain a grade of A or B, or close until passing a future inspection.” Figure 1 displays the distribution of citations across the different boroughs for January 2019 through Feb 2020, with approximately 87.4% of locations maintain a high-quality rating, with only 5% receiving a grade of C. There are several outliers, such as Dunkan Donuts is consistently the most infractions, followed by McDonalds and Star Bucks, in part due to the number of locations. Specific ethnic categories, such as Chinese in Queens and Indian in the Bronx, have high infractions due to their concentration. Niche categories, such as Cajun-Creole, have an alarming rate of 22% non-compliance due to their only being 58 total businesses in the category.

Figure Citations by Borough



The feed also details the violation code for the citation along with a numeric score that typically has a mean value of 20.4 and a standard deviation of 14.8 and a critical flag, with lower scores representing more minor infractions. A goal of the research is to predict the score that an organization would receive. First, the features Cuisine Description, Critical Flag, Violation Code, Inspection Type were One Hot Encoded into a 586 by 200730 matrix and fed into Scikit Learn’s Logistic Regression algorithm using a 50% split for testing/training data. The test data accuracy was 10.8%, making this strategy unusable. Next, the Borough, Critical Flag, Grade, Violation Code, and Inspection Type was One Hot Encoded into a 75 x 200730 matrix for the Decision Tree Regressor algorithm. This solution has an accuracy of 67.8% and was took significantly less time to train. Next, a Principal Components Analysis (PCA) over this same matrix was able to reduce the matrix dimensionality to 25 x 200730. This result did not change the accuracy results of either algorithm though it did speed up the training by 2-3x. Then the score was partitioned into value ranges -5 to 15, 16 to 25, 26 to 50, and 50 to 200. Using the Decision Tree Classifier was able to predict the bucket with a 93.8% accuracy. Additional analysis found that the Grade feature has the strongest influence, and its removal decreases the Decision Tree Classifier’s accuracy to only 51.7%. Removing both the Grade and PCA preprocessing increases the One Hot Encoded matrix from 10 x 190803 to 68 x 190803 with a total accuracy of 0.03%. This test suggests that there is insufficient data for this strategy, and the boost from using PCA correlations is critical for the sparse data set. Finally, MLPRegressor, a neural network-based algorithm was tested to have a 64.8% accuracy when both the Grade and PCA are available. After some initial effort to prepare the data Scikit follows a very consistent pattern that makes it easy to change the algorithm and iterate quickly.

Table : Machine Learning Algorithms Used

|  |  |  |
| --- | --- | --- |
| Algorithm | Strategy Description | Comments |
| Logistic Regression | Maps numeric features to a numeric range. | The label-centric data set was not very applicable |
| Decision Tree Regression | Maps vectorized labels to a numeric range. | Effective with many guesses in general ballpark |
| PCA | Uses the correlation between features and combine columns | Improves runtime and accuracy of sparse data |
| Decision Tree Classifier | Maps vectorized labels to finite categories (e.g., value-ranges) | More accurate than DTR as general guesses are sufficient |
| MLPRegressor | Uses backpropagation to build a fitting expression | Suffers from the same challenges as LR |

# Section II: Analyze the Process