Predicting Permit Processing Times for Corporations in Brooklyn

Yash Deshpande

April 2, 2019

1 Introduction

New York City's Department of Buildings (DOB) issues permits for construction and demolition activities in the City of New York. Using their permit issuance data-set, I attempt to compare the efficacy of a combination of feature pre-processing and predictive modeling techniques in predicting the processing time (in days) for corporations in Brooklyn.

2 The Data: Overview and Exploration

What does the data look like? Each row/record in this data-set represents the life cycle of one permit for one work type. The data is updated daily, and goes back 30 years. For this analysis, data was pulled on **March 3rd, 2019**. The full data contains 60 features, such as:

- Borough (Manhattan, Brooklyn, etc.);
- Job Type (Job type codes as per the DOB- new building, alteration, etc.);
- Special District 1, Special District 2 (Indicators for 2 different special district categories);
- Owner's Business Type (Individual, Corporation, etc.);
- etc.

For the purposes of this assignment, I chose to restrict the data to Initial permits issued to corporation addresses in the borough of Brooklyn. There was a small subset of this data that had negative Processing Times (≈ 97)- these were dropped before further analysis.

Some exploratory visualizations

Figures 2.1 - 2.3 The distribution of permit processing time in days for Corporations in Brooklyn. The processing time in the data exhibits a mean value of 3.201 days, and a standard deviation of 30.330 days. As seen in *Figure 2.3*, the number of permits with higher processing times drops considerably, especially after crossing approx. 150 days.

FIGURE 2.2 Variations in the total number of permits issued to CORPORATION addresses in BROOKLYN, grouped by PERMIT TYPE into 5-year time buckets.

FIGURE 2.3 Variations in the average processing time for permits issued to CORPORATION addresses in BROOKLYN, grouped by PERMIT TYPE into 5-year time buckets.

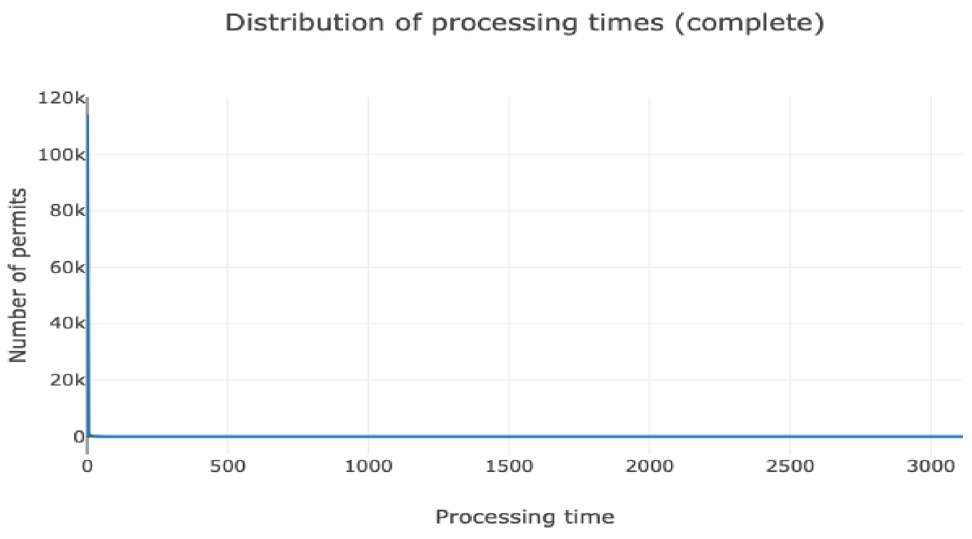

Figure 2.1: Distribution- complete

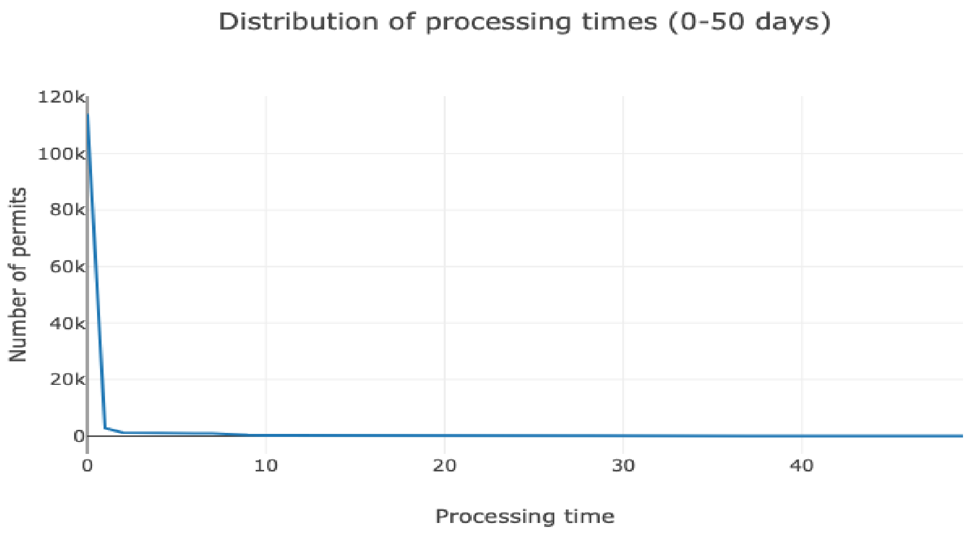

Figure 2.2: Distribution- 0 to 50 days

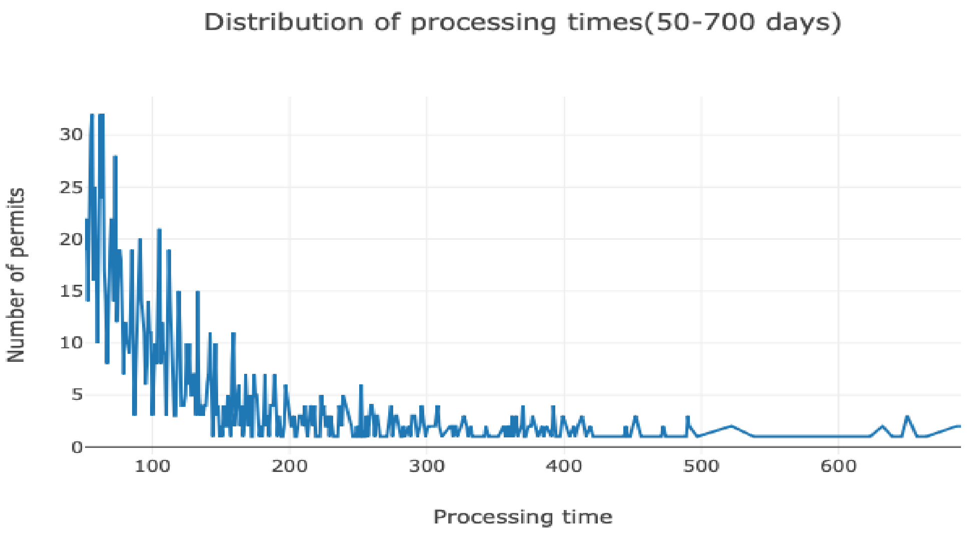

Figure 2.3: Distribution- 50 to 700 days

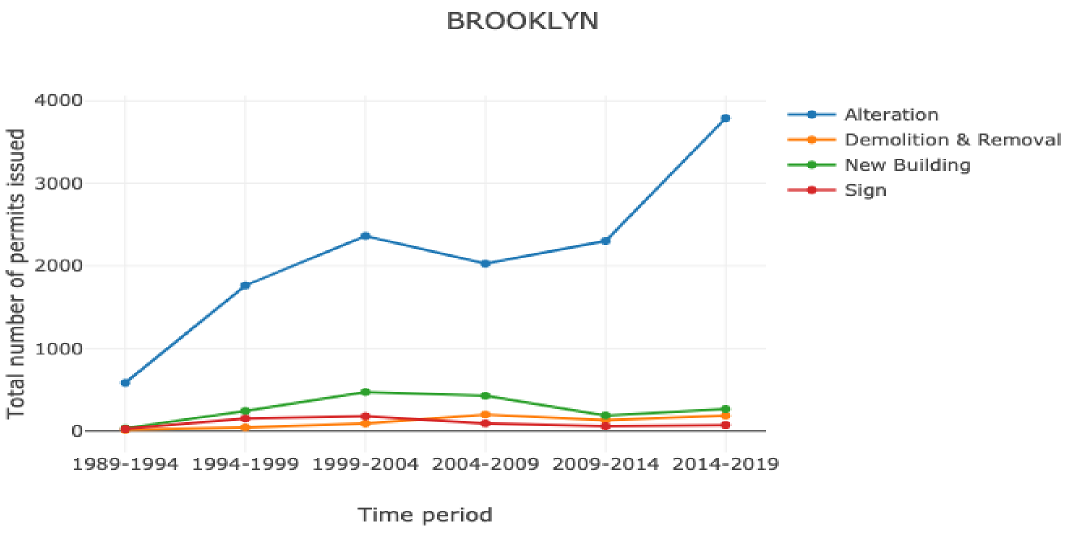

Figure 2.4: Total number of corporation permits issued

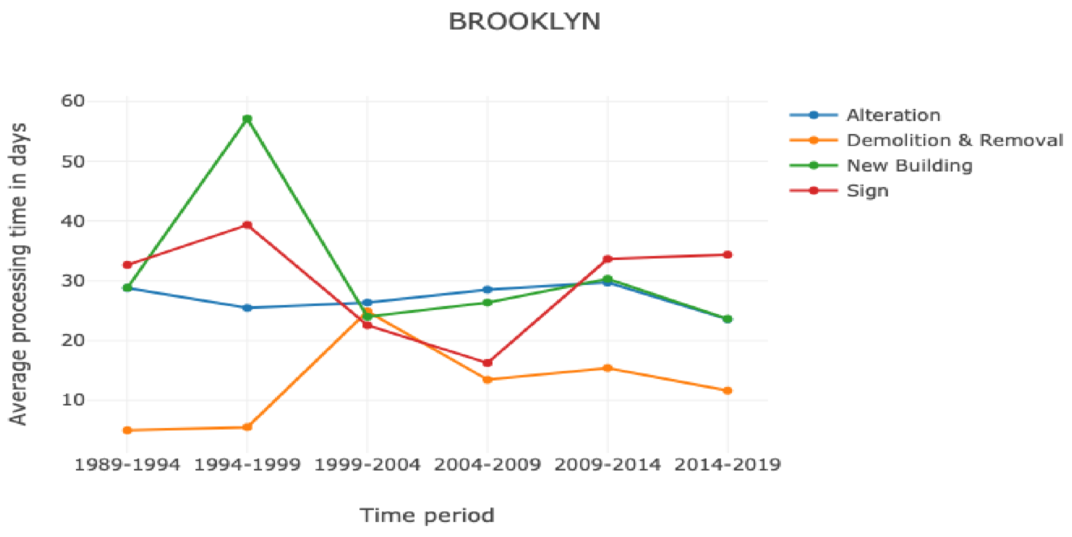

Figure 2.5: Average processing time for corporation permits

3 Data pre-processing

The Processing Time for each permit in days was inferred from the Filing Date and Issuance Date columns. Renewals were not considered. The features retained were all categorical variables, and are listed below along with a brief description. Their choice was affected by my own interest in deducing their predictive value for the problem at hand.

- Special District 1: which designated special district (specific) a permit address falls under (e.g. Bay Ridge)
- Special District 2: which designated special district (generic) a permit address falls under (e.g. Industrial business zones)
- Non-Profit: whether or not the corporation is a non-profit organization (binary)
- Permittee's License Type: the license type of the agent filing for the permit
- Permit Type: the type of permit issued
- Residential: whether or not the corporation
- Bldg Type: the type of building involved
- Work Type: the type of work involved (e.g. plumbing)
- Processing Time: **to be predicted!**

Feature extraction   Two types of feature extraction techniques were employed and compared, and have been listed below. Both techniques employ black-box Python pre-processors from the scikit-learn and Gensim packages. A Train/Test split of 70%/30% was employed.

1. Label encoding followed by one-hot encoding
    - Each feature was converted to a string, then encoded as a label. This maps n distinct categories to n distinct integer values. This was done prior to one-hot encoding, as the one-hot encoder implicitly ignores unknown labels.
    - Post label encoding, a one-hot encoder was used to transform the data to a sparse matrix representation, which is then fed to the predictive models as a featurized matrix.

2. WORD VECTORIZATION
    - WORD2VEC has proven to be a powerful set of word embeddings in NLP applications. In this case, each row of the data can be thought of as a set of different labels that are analogous to a *sentence*.
    - Each *sentence* is then learned from the training data as a vector representation, which is then fed to the predictive models as a featurized matrix.

4 PREDICTIVE MODELING

MODELS EMPLOYED Three models were employed, with a 3-fold cross-validated GRID SEARCH for parameter tuning. These models are listed below, along with the respective range of parameters tested.

1. LASSO REGRESSION
    - Regularization weight, $\lambda \in \{10^{-4}, 10^{-3}, 0.01, 0.06, 0.1, 0.3, 0.5\}$
2. RIDGE REGRESSION
    - Regularization weight, $\lambda \in \{10^{-4}, 10^{-3}, 0.01, 0.06, 0.1, 0.3, 0.5\}$
3. Random Forest Regression
    - Number of trees $\in \{25, 40, 55\}$
    - Maximum depth $\in \{10, 20, 30\}$

RESULTS *Mean absolute error* (unit: $days$) was the metric used for model evaluation. The details of the best performing model configurations are tabulated below for each featurization method.

Model	**Parameters**	**Train**	**Validation**	**Test**
Lasso	$\lambda = 0.01$	5.61 d	5.61 d	5.81 d
Ridge	$\lambda = 0.0001$	5.61 d	5.62 d	5.81 d
Random forest	$n_{trees} = 40$; $max.\ depth = 10$	5.55 d	5.60 d	5.80 d

Table 4.1: Results- one-hot encoded featurization

Model	**Parameters**	**Train**	**Validation**	**Test**
Lasso	$\lambda = 0.1$	5.82 d	5.82 d	5.93 d
Ridge	$\lambda = 0.5$	5.82 d	5.83 d	5.93 d
Random forest	$n_{trees} = 55$; $max.\ depth = 10$	5.45 d	5.90 d	6.01 d

Table 4.2: Results- Word2Vec featurization

Evaluating model predictions A minimum mean absolute error of 5.6 days was achieved on the test set with one-hot encoded features as input to a random forest model with $n_{trees} = 40$ and $max.\ depth = 10$ (highlighted above in *Table 4.1*).

Figure 4.1 below compares the model's predictions to the actual processing time in days for the first 100 samples in the test set. As seen, although the mean absolute error is small compared to the standard deviation of processing time in the actual data, the predictions are not very effective. Outliers in the data stand out; the model seems to predict processing time only within a small range of values.

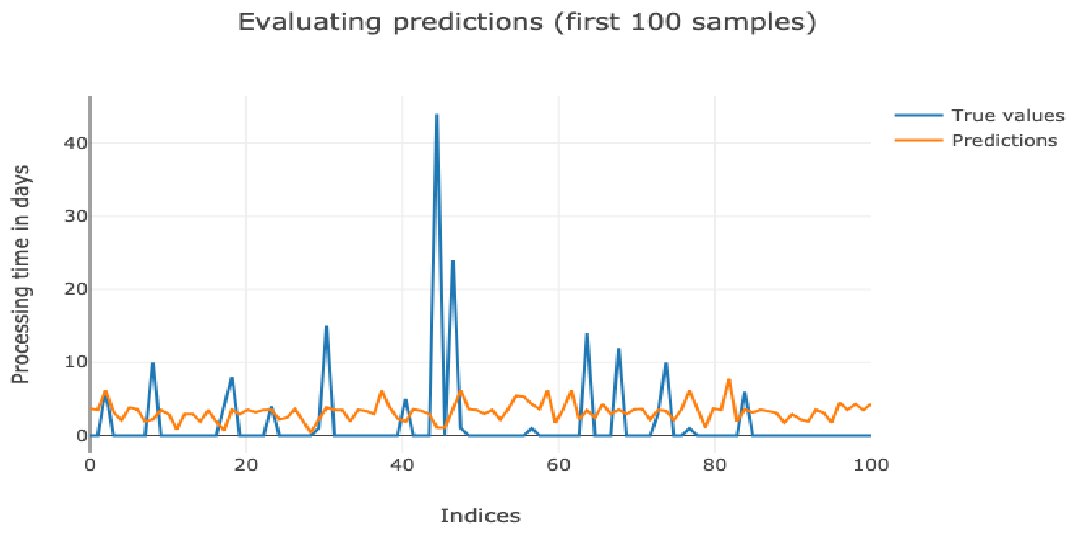

Figure 4.1: Comparing first 100 predictions

5 Conclusions and Further Scope

The predictive model of choice did not perform well. There are many explanations to this, the most likely of which are:

- The choice of features was not very indicative of the dependent variable.
- The nature of data itself (all features) was not conducive to a predictive modeling setup.

Further Scope Additional avenues that can be explored in order to potentially improve the efficacy of the model include:

- Contextualizing the predictive problem using other data sets, e.g. demographic and population data, Department of Buildings employment data (number of employees by district, etc.).
- Attempting to featurize categorical variables using neural networks- entity embeddings, as described by Cheng Guo and Felix Berkhahn in their 2016 paper, Entity Embeddings of Categorical Variables.

6 CODE

The code for this assignment can be found, bucketed as per the sections of this report, in the following GitHub repository: `https://github.com/yashd94/ToposAssignment`.