Denver Restaurant Inspections

James Howard james.howard@jhuapl.edu
Alicia Bargar alicia.bargar@jhuapl.edu

January 30, 2017

Code

All code has been committed to `restaurant_inspections/Denver/code` in the GitHub repository. The file `DenverDataPrep.Rmd` was used to prepare the data for analysis.

Data Set

Data Sources

Enforcement of health codes at food service establishments (FSE) in Denver, Colorado, is the responsibility of the City and County of Denver Department of Environmental Health, and all data acquired during health code inspections are cataloged by the city for enforcing health regulations. The data set was composed of all food service inspections performed in Denver between 2 January 2013 and 4 November 2016.

Violation Types

Food service establishments (FSE) are regularly inspected by Denver as part of the health regulations. FSEs may receive one or more violations for different types of infractions. Overall, there are two categories of infractions for FSEs, critical and non-critical violations. When the county cites an FSE with a critical violation, the FSE must correct it immediately. If the violation is not immediately addressed, the county can force the FSE to close in order to remediate the violation since it is considered to be a public health hazard. When the county cites an FSE with a non-critical violation, it must be remediated within a time frame set by the inspector. The FSE may continue to operate during this time frame since the violation is not considered to be evidence of an imminent public health hazard. The types of infractions, categorized by criticality, are provided in Table 1.

Table 1: Violation types

Critical Violations	Non-Critical Violations
Unwholesome; signs of spoilage	Refrigeration units not provided with accurate, conspicuous thermometer
Cross-contamination	Dish machine not provided with accurate thermometer and gauge cock
Personnel with infections not restricted	Chemical test kits not provided; inaccessible
Wounds unprotected	Dishwashing operations
Hands not washed as needed	Utensils not provided; used/stored improperly
Smoking; eating; drinking not restricted	Single service articles improperly stored, dispensed, used
Training needed	Reuse of single service articles

Critical Violations	Non-Critical Violations
Bare hand contact	Plumbing not installed/maintained
Food thermometer not available	Garbage and refuse accumulation/uncovered
Equipment inadequate to maintain food temperatures	Floors; walls; ceilings in disrepair
Unsafe water source	Lighting inadequate
Hot & cold water inadequate	Ventilation inadequate
Improper sewage disposal	Personnel; unauthorized; unclean clothes; hair unrestrained
Soap and drying devices unavailable	Linen improperly stored
Evidence of insects or rodents	
Inappropriate pesticide application	
Evidence of animals on the premises	
Food unprotected from contamination	

Food Service Establishments

The data set was originally provided as a table with each inspection corresponding to a single row. Each inspection includes a binary variable representing whether or not a violation of each type was found. In this data set, each violation may include multiple subcategories that correspond to whether or not a box was checked on an original violation on a report form. The total number of inspections is $n =$ '26781', and for each inspection, the number of major violations and minor violations was calculated by counting the number of TRUE responses for each class of violation.

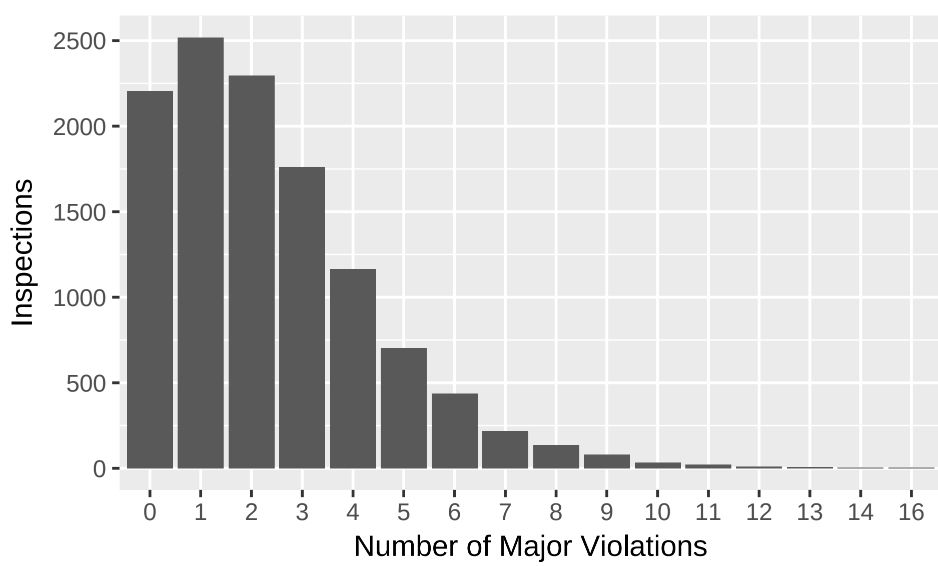

Figure 1: Number of Major Violations per Inspection

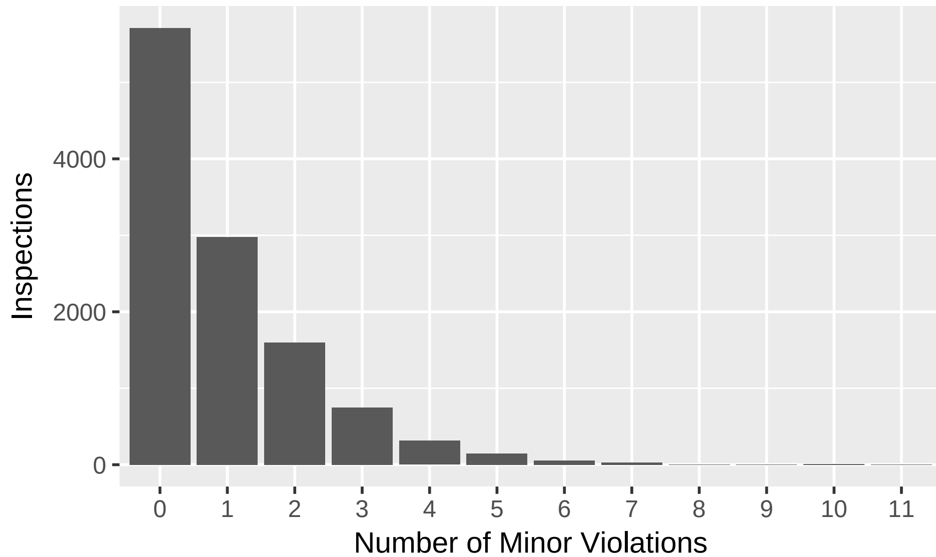

Figure 2: Number of Minor Violations per Inspection

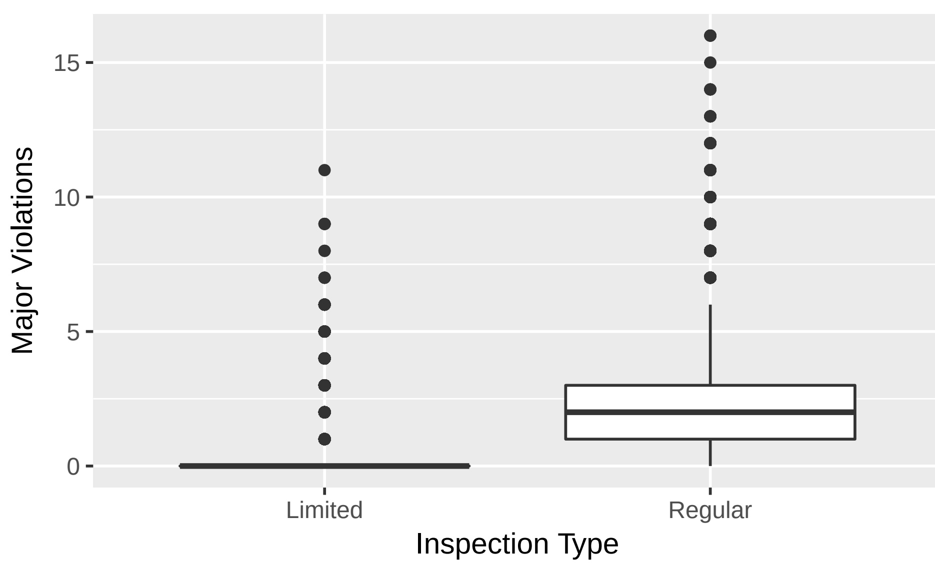

Figure 3: Number of Major Violations by Inspection Type

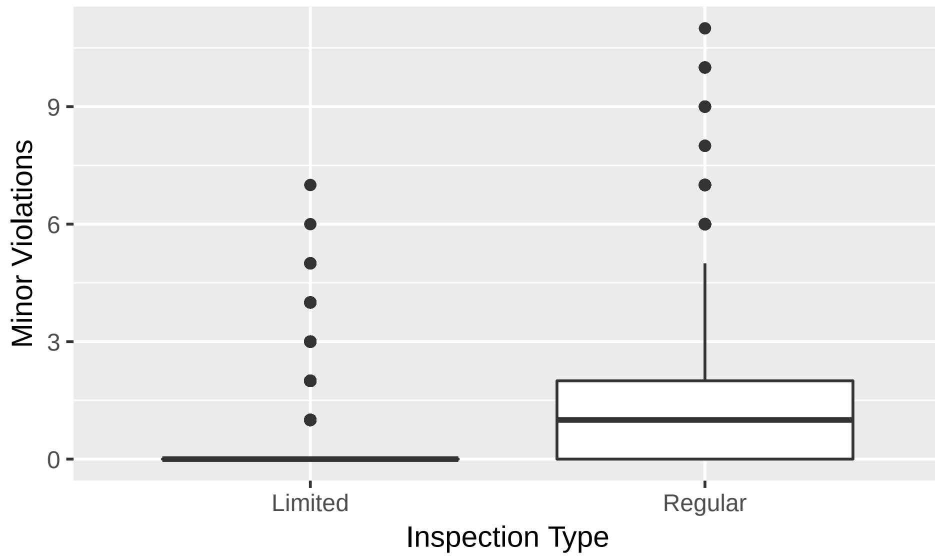

Figure 4: Number of Minor Violations by Inspection Type

Inspection Type

Of the 20987 inspections, 12865 were regular inspections and 8122 were limited inspections. Limited inspections are scheduled in response to a previously found violation. Figures 3 and 4 show how the number of critical and non-critical violations are distributed according to inspection type. The number of violations (critical or non-critical) tends to be lower for reinspections, since the re-inspection is being performed to verify those specific violations have been remediated. The difference was determined to be statistically significant ($p < 0.01$) via a t-test for both critical and non-critical violations.

FSE Category

There are some inspection actions for which there is no corresponding restaurant data. Of those, we have full establishment data on 11595 inspections. Of those 11595 inspections, 11443 were restaurant inspections and 152 were non-restaurants, such as school cafeterias, and other institutional food service outlets. Due to this severe imbalance in types, all non-restaurants were removed from both the training and test data.

Methodology

Data Preparation

Following the example set by the Chicago study, we sought to develop a statistical model that would produce a binary prediction of whether an inspection would yield at least one critical violation. Since critical violations are considered to pose a public health risk, the ability to predict whether inspecting a particular FSE would yield at least one violation can allow health inspectors to prioritize higher-risk FSEs over lower-risk ones when scheduling inspections. Since the predictor is binary, its performance can be assessed using conventional metrics like the receiver operating characteristic (ROC) curve, and area under the curve (AUC).

In order to fairly evaluate the models, the data was divided into training and testing sets. The training data consisted of all inspections in the dataset before 31 December 2015. The testing data consisted of all inspections from 1 January 2016 onward. Only regular inspections were considered for training and testing (re-inspections were removed). The number of training instances is `4185` and the number of test instances is `2598`.

Model

For our baseline, we apply a basic model based on prior information and inspection history and evolve our model from there.

Baseline Model

A convenient baseline approach bases the prediction off of one indicator, the number of critical violations from the previous inspection. This approach expects that restaurants with a history of health code violations will continue to exhibit similar behavior. The model was implemented using logistic regression via the `glm` function in R and outputs the posterior probability of an inspection having at least one critical violation.

Alternative Model 1

A variety of additional features were considered for enhancing the baseline model. Several of these features were also chosen for the Chicago model. These features include:

- Number of major violations in the last inspection
- Number of minor violations in the last inspection
- Days since the last inspection
- Average number of neighboring major violations
- Average number of neighboring minor violations

Alternative Model 2

We also include a model with an indicator variable for whether the previous inspector is the inspector for the current inspection. This model is meant to test the correlation between inspector id and inspection result based on insights from the Chicago and Raleigh models.

- Number of major violations in the last inspection
- Number of minor violations in the last inspection
- Whether the last inspector was the same as current

Results

We use logistic regression to implement all three models to determine their effectiveness and we use the area under the curve (AUC) as the metric for evaluating model efficiency. The AUC measures how likely a model will rate a random truly positive example versus a random truly negative example. The base measure, equivalent to flipping a coin with no outside information, is 0.5. A perfect model would yield an AUC of 1. We use the test dataset, described above to provide examples for testing. The results of the AUC are given in Table 2 and the plot of resulting curves is shown in Figure 5.

In addition, we review the same three model inputs with respect to the RandomForest algorithm. We use RandomForest as it has rapidly become a staple of machine learning and generally performs well on diverse datasets. For our analysis, the corresponding AUC for RandomForest are shown in Table 2 and the plot is given in Figure 6.

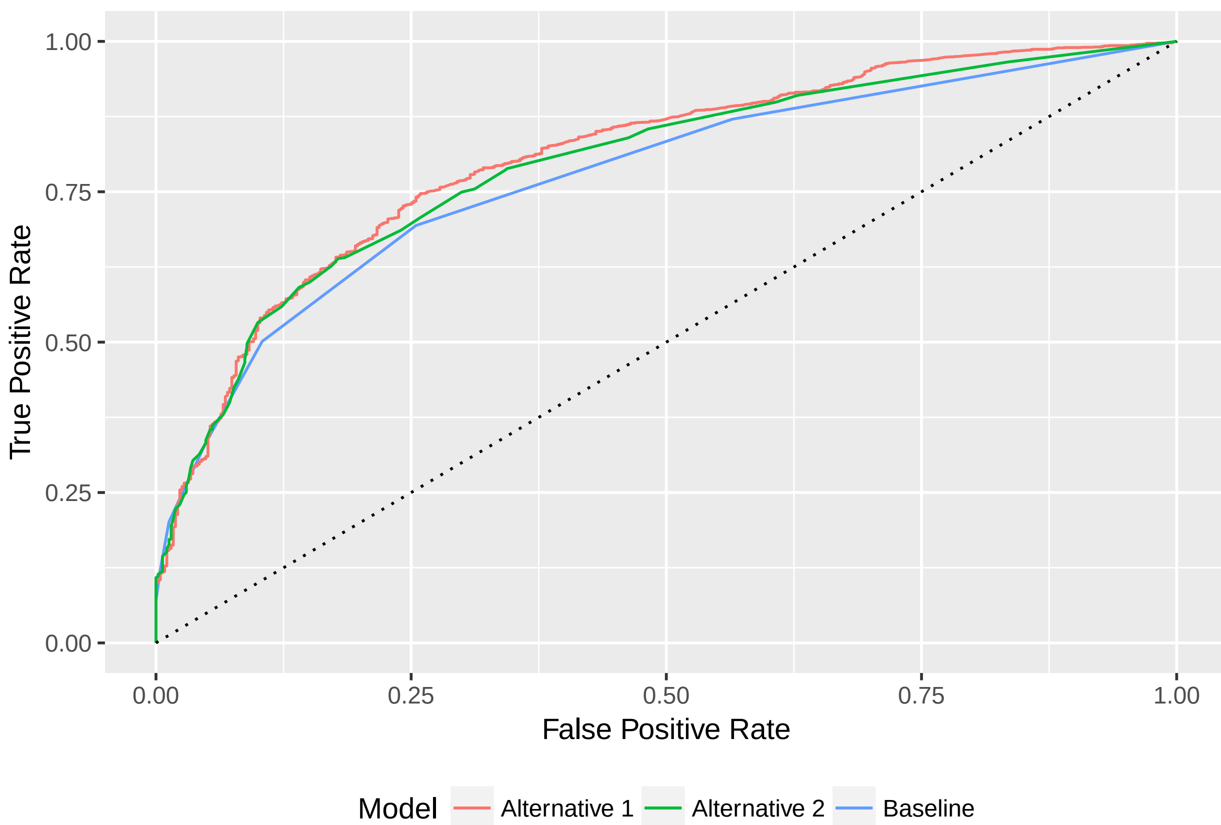

Figure 5: ROC Curves for Logistic Regression Models

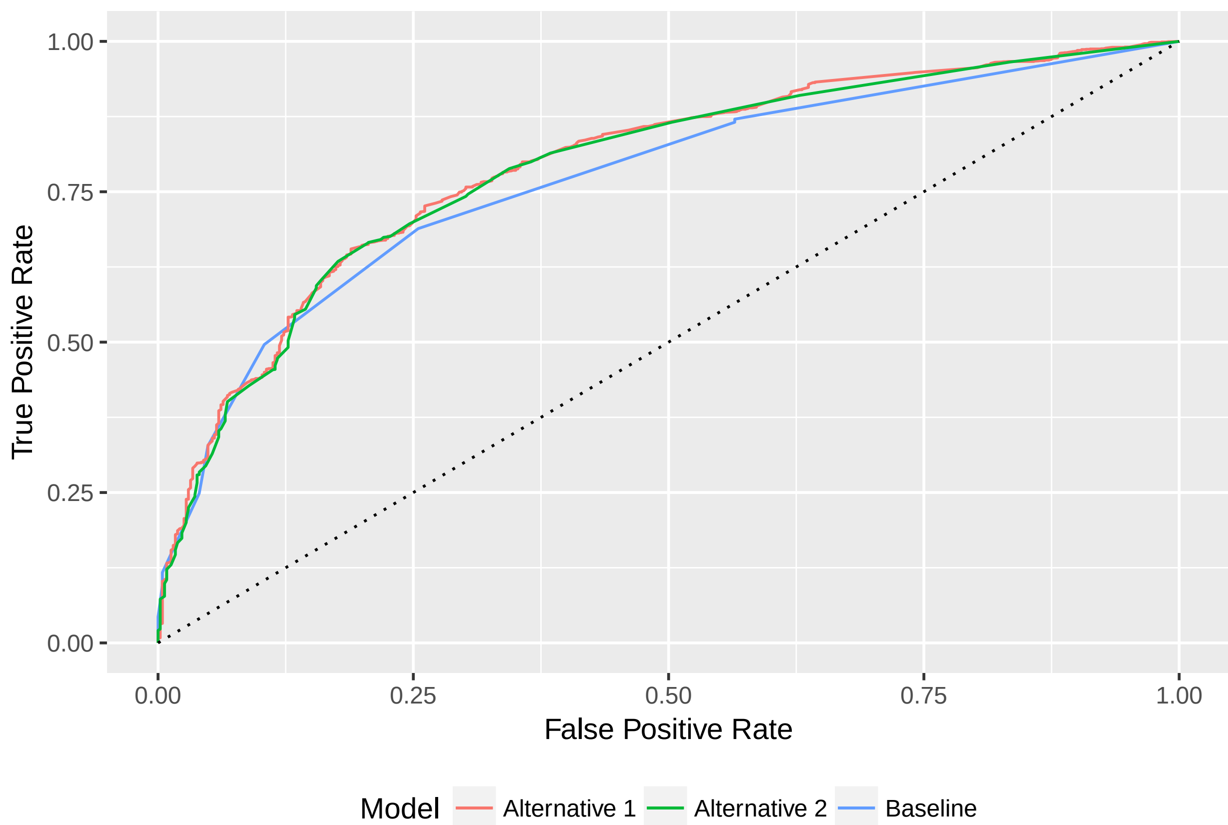

Figure 6: ROC Curves for RandomForest Models

Table 2: Model Performance

Model	AUC (Logistic)	Days Saved	AUC (RandomForest)	Days Saved
Baseline	0.7741289	2.7227564	0.7693835	2.6842949
Alternative 1	0.8066618	3.0304487	0.7933824	3.025641
Alternative 2	0.7941475	2.9967949	0.7883131	2.9198718

The other performance metric that may be of more relevant to local health departments is the number of days earlier (or later) in which critical violations were found, compared to a "business as usual" inspection scheduling. To compute this metric, the classifier output scores are sorted in descending order and then associated with the original dates of the test inspections. It is assumed that inspections would have occurred on the same dates, with the same number of inspections per day, but the order in which FSEs are inspected is determined by the data-driven model. The inspection dates for the new schedule are compared to the dates of the old schedule, and the mean difference in inspection date is computed. A positive number indicates that FSEs with critical violations are being inspected earlier, and a negative number indicates that they are being inspected later. Sorting the list randomly would result in zero days saved on average. The mean number of days saved is shown in Table 3.

Table 3: Model Performance

Model	Days Saved (Logistic)	Days Saved (RandomForest)
Baseline	2.7227564	2.6842949
Alternative 1	3.0304487	3.025641
Alternative 2	2.9967949	2.9198718

Inspector Influence

The work on Raleigh's data found inspector assignment to be highly informative of a restaurant's performance. To test whether this was true for Denver as well, we ran our model through a similar procedure to control for inspector influence. Note that we did not have inspector data for all investigations, so our training and testing sets were reduced accordingly: from $N_{\text{train}} = 5353$ to $N_{\text{train\_new}} = 4185$ and $N_{\text{test}} = 2751$ to $N_{\text{test\_new}} = 2652$. First, we trained an inspector-only model to compare the value of this variable to those used in our other models. The AUC for the inspector-only model was 0.584, compared to an AUC of 0.800 for the original logistical model over the same data. This indicates our model captures information beyond what the inspector variable yields. This is distinct from Raleigh's outcome, where the inspector-only model gave comparable performance.

Next, we checked how often a restaurant was inspected by the same inspector twice in a row. We found that this was true for 43.7% of our cases. We subset our data to only cases where the second inspection was performed by a different inspector, yielding new set sizes $N_{\text{train\_new2}} = 2525$ and $N_{\text{test\_new2}} = 1322$. On this reduced set, our AUC performance was 0.772. As a result, we are able to confirm that the model's improved accuracy is due to information apart from inspector influence.

Discussion

In this working paper, we have attempted to apply the Chicago restaurant inspection model to data provided by the City of Denver with an objective of improving the restaurant inspection process. We have developed models that focus, primarily, on the last inspection results finding those results are the best indicator of whether a restaurant will have a violation reported in the current inspection. Additionally important is the most recent violation history of other nearby restaurant facilities. However, using this information, we are able to reschedule restaurant inspections to cut the number of days an FSE is out of compliance by at least three days in the average case.