Memo for Raleigh Work

Author: Alex Perrone.

Updated: 01/13/17.

Code

All code has been committed to `restaurant_inspections/raleigh/code`. The file `merged.csv` represents all the cleaned and merged data from inspections/violations, Wake county restaurant information, Census data, and Yelp data. The file `pres-Nov17.Rnw` was used to generate the slides `pres-Nov17.pdf` which were presented to GovEx on 17 November 2016.

Dataset

The workflow for Raleigh is as follows. Each node represents a script (such as an R or Python script) where red indicates the input files and blue indicates output files. Unless mentioned otherwise, code is in `raleigh/code` and data are in `raleigh/data`.

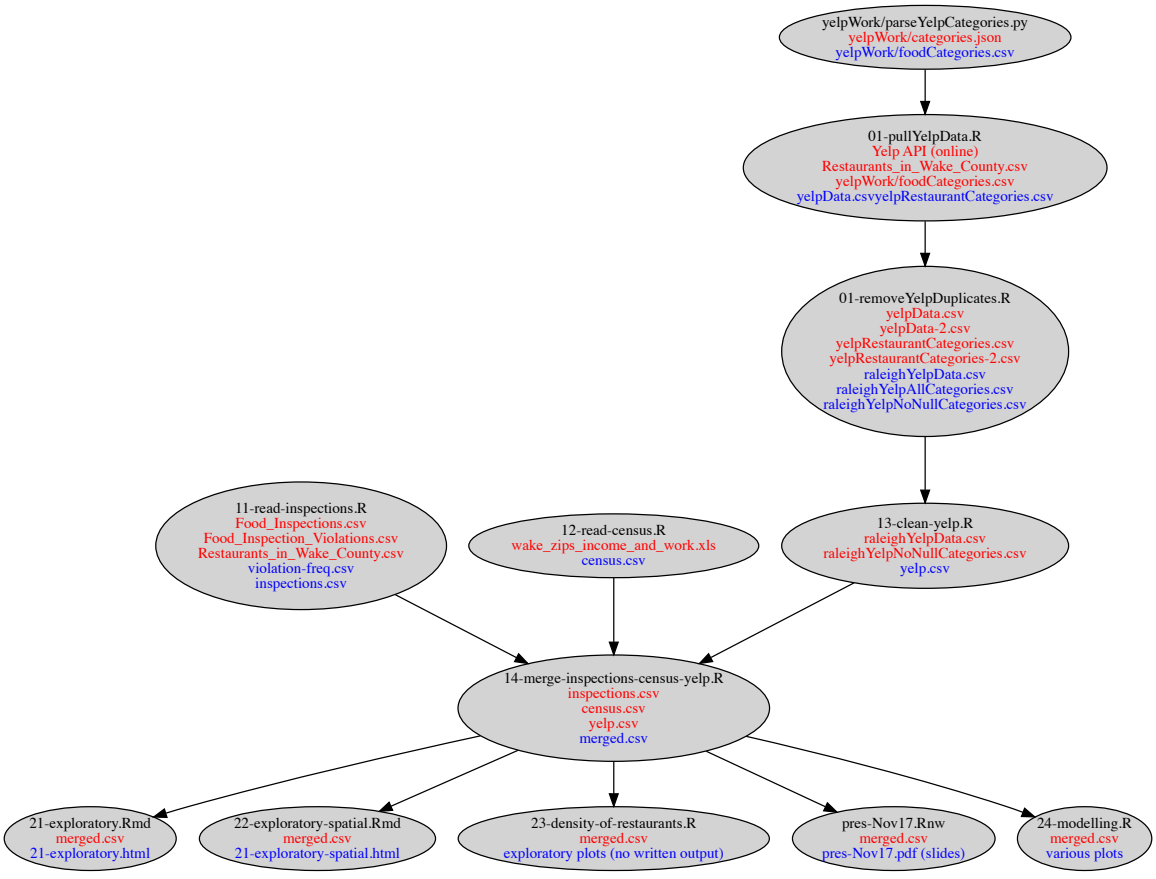


Figure 1: Pipeline for Raleigh restaurant inspections.

The image can be found in `restaurant_inspections/raleigh/doc/workflow.pdf`, which was generated by its corresponding `workflow.dot`.

Inspections/Violations

Relevant file: `11-read-inspections.R`.

Inspections and violations data were downloaded from the Open Data website for Wake County, North Carolina at http://data.wake.opendata.arcgis.com/datasets?q=Restaurant. The data included all food service inspections in Wake County between 21 September 2012 and 03 November 2016. The inspections (`Food_Inspections.csv`) and violations (`Food_Inspection_Violations.csv`) were in two separate files with a common key `HSISID`. Since these files included no restaurant information such as name, address, zip, facility type, etc., the file `Restaurants_in_Wake_County.csv` was used to provide this information downloaded from the same website above; this file also had the key `HSISID`. The inspections included Re-Inspections (approximately 1.5% of the data) and these were excluded from analysis.

Violation codes along with their descriptions can be found from the Wake County manual at http://www.wakegov.com/food/healthinspections/rules/Documents/NC%20Food%20Code%20Manual%202009%20FINAL.pdf. The violations file was provided in the format of one violation per row, thus a given inspection may span multiple rows. However, each violation code reported for an inspection was listed as critical ("Yes") or not ("No" or NA). The violations were aggregated to counts of critical and non-critical violations per inspection using whether they were listed as "Yes" or not in the critical column.

There were 299 unique violation codes found in the data. The top 10 most common violations as well as their percentage occurence are given in the following table. Percentages indicate the fraction of inspections which reported the given violation code.

Code	Short Description	Frequency
4-501.11	Good Repair and Proper Adjustment of Equipment	0.420
6-501.12	Cleaning of Physical Facilities	0.321
3-501.17	Ready-to-Eat, Potentially Hazardous Food (Time/Temperature Control for Safety Food), Date Marking	0.263
4-601.11(B)(C)	(B) Food contact surfaces clean, (C) Non-Food-Contact surfaces free of dust, dirt, food residue	0.305
4-601.11(A)	Equipment food-contact surfaces and utensiles clean	0.253
3-501.16(A)(2)(B)	Potentially hazardous food kept at proper temperature	0.235
2-102.12	Supervisor is ANSI-certified as food protection manager	0.245
3-304.14	Proper use of wiping cloths	0.232
4-602.13	Non-Food-Contact surfaces cleaned at frequency to prevent residue	0.234
3-302.11	Food protected by cross-contamination	0.178

The data sources (inspections, violations, restaurant info) were inner-joined by `HSISID` such that any records that did not match across all three data sources were excluded. In particular, the inspections and violations datasets were joined by `HSISID` and `Date` so that the inspections and violations matched the facility in time, then this joined dataset was matched with the restaurant information by `HSISID`.

Dataset	Number of Rows	Number of Unique HSISID
Inspections	23860	4569
Violations	22968	4270
Restaurant Information	3324	3324
Merged Inspections	18466	3045

Each inspection resulted in a Score, but number of critical violations were instead used as a target (response) variable, in line with the Chicago evaluation. A histogram of the number of critical violations in the Raleigh data is shown in the following plot, which reveals that in comparison to the Chicago data where most inspections resulted in 0 or 1 critical violations, the Raleigh data has more critical violations with a larger spread of values.

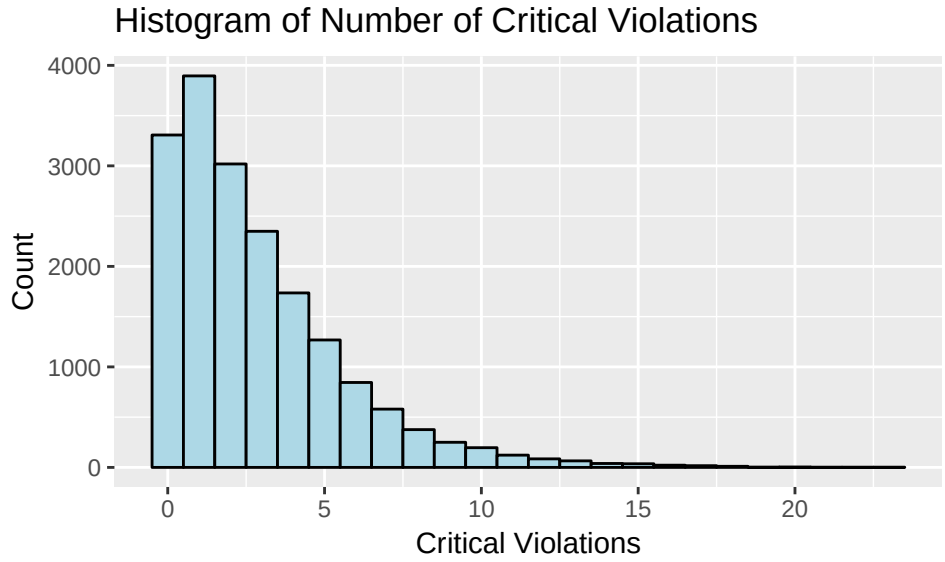


Figure 2: Variable importance for Random Forest Classification.

There were numerous categories of facility types, as shown in table below. All types were included, though many of these types were excluded in modeling because they did not match with Yelp data, and due to time constraints, only complete cases were used instead of creating models with different sources of data, as was done in Food Inspection Violation, Anticipating Risk report for Montgomery County, MD at http://www.mayorsinnovation.org/images/uploads/pdf/1_-_Montgomery_MD.pdf.

FacilityType	N
Restaurant	12086
Food Stand	3199
Public School Lunchrooms	1587
Meat Market	732
Institutional Food Service	352
Mobile Food Units	195
Private School Lunchrooms	135
Pushcarts	111
Elderly Sites	69
Limited Food Service	3

Census

Relevant files: `12-read-census.R`, `14-merge-inspections-census-yelp.R`.

There were six census data sources, each at the ZIP code level:

- Income and work
- Demographics
- Race
- Foreign-born characteristics
- Occupancy
- Property tax

Of these, only the Income and work information was used, in particular for the median household income and the percent below poverty line. The census data were merged with the inspections data, which reduced the inspections from 18,466 rows to 18,231 rows.

Yelp data

Relevant files: `13-clean-yelp.R`, `14-merge-inspections-census-yelp.R`.

The Yelp data were merged with the inspections data using the phone number. This resulted in only about 2/3 of the rows matching. Efforts were made to approximately match by restaurant name using several methods (Levenshtein, restricted Damerau-Leveshtein) with different maximum distance parameters. However, manual inspection revealed that restaurants that matched using the approximate string matching yet whose phone numbers differ (and thus would not be matched by joining on phone number) were almost universally false positives as they differed substantially in other fields such as their address. A better approach that approximately matches on both name and address might yield better results.

The Yelp data included many restaurant categories. Somewhat arbitrarily, the top 20 restaurant categories were used. The frequencies of each of the top 20 most frequent categories are shown in the following table. Note that categories are not exclusive, as a facility may be fall in several categories, thus the frequency column does not sum to 1.

Yelp Category	Frequency
grocery	0.177
tradamerican	0.107
sandwiches	0.105
hotdogs	0.102
pizza	0.089
mexican	0.074
burgers	0.065
chinese	0.062
chicken_wings	0.052
breakfast_brunch	0.049
italian	0.047
delis	0.040
newamerican	0.034
sushi	0.034
bars	0.032
salad	0.029
coffee	0.029
seafood	0.026
bbq	0.023
bakeries	0.012

The Yelp matching reduced the inspections data from 18,231 to 12,069 rows.

Density of restaurants

It was hypothesized that the density of restaurants (computed as number of restaurants within a fixed radius of given restaurant) might be important for predicting the number of critical violations. This was computed using various settings for the radius parameter of 0.5km, 1km, 2km, and 3km. However, exploratory plots of number of critical violations vs. restaurant density revealed practically zero relationship, so this variable was not used (see `raleigh/code/23-density-of-restaurants.R` for further details).

Summary of data sources

A complete summary of merging the data sources is provided in the following table.

Dataset	Number of Rows	Number of Unique HSISID
Inspections	23860	4569
Violations	22968	4270
Restaurant Information	3324	3324
Merged Inspections	18466	3045
Merged Inspections + Census	18231	2999
Merged Inspections + Census + Yelp	12069	1998
Merged Inspections + Census + Yelp (Complete Cases)	9265	1710

Modelling

Relevant file: `24-modelling.R`. The following variables were included in the model:

- Inspection/Violations
  - Inspection resulted in at least one critical violation (binary target variable)
  - Number of critical violations (continuous target variable for secondary models)
  - Number of previous critical violations (continuous variable)
  - Number of previous non-critical violations (continuous variable)
  - Mean number of all previous critical violations (continuous variable)
  - Mean number of all previous non-critical violations (continuous variable)
  - Days from last inspection (continuous variable)
  - Days since opening (continuous variable)
  - Facility Type (categorical variable)
- Geographical Information
  - Average number of critical violations for all prior inspections of nearest 5 neighbors (continuous variable)
- Census data: extracted income information by ZIP
  - Median household income (continuous variable)
  - Percent below poverty line (continuous on [0, 1])
- Yelp data: extracted information on restaurants
  - Rating (out of 5 stars)
  - Price ($-$$$$)
  - Restaurant category (top 20, e.g. Mexican, Sushi, Chinese, Coffee)

Unlike the Chicago evaluation, inspector information was not included in the model. Zip code was also not included in favor of ZIP-level census information.

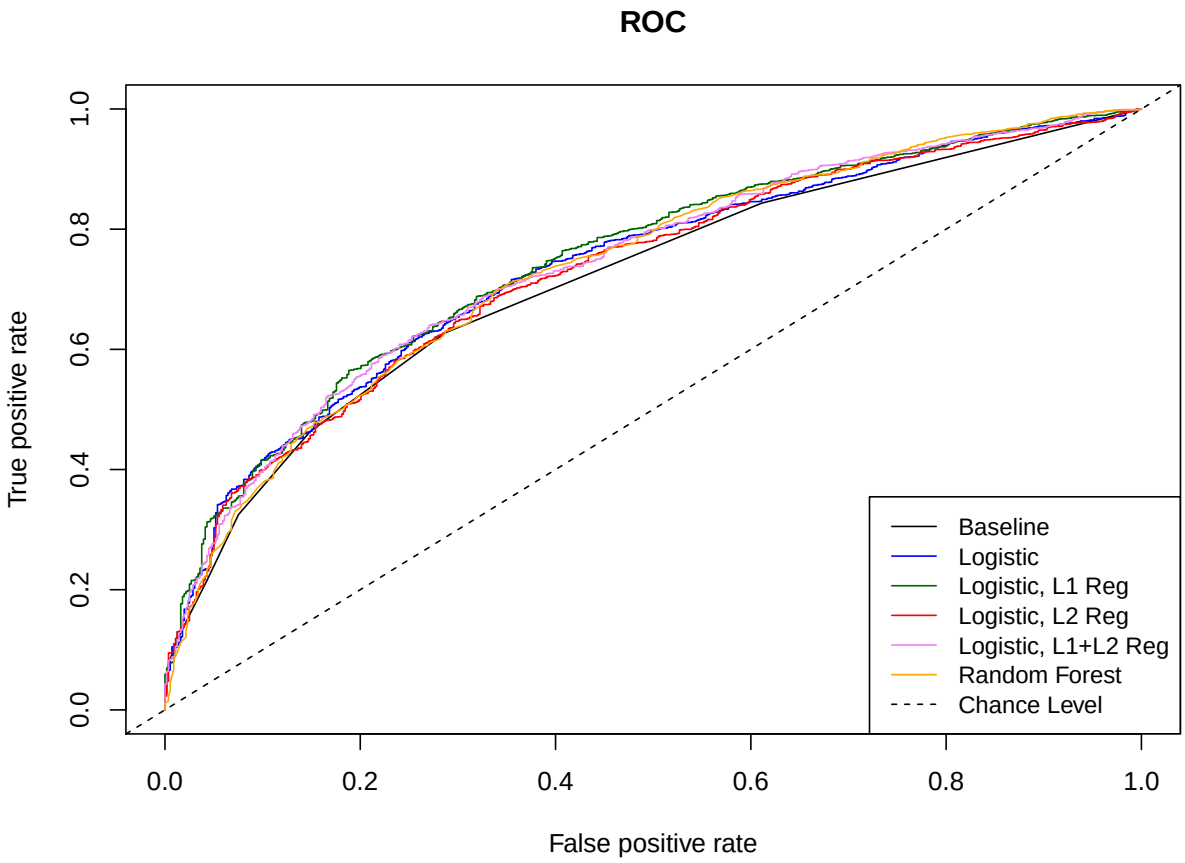
There were various aspects of the data that were missing, mostly those facilities which did not have Yelp

reviews. Due to time constraints, only complete records were analyzed. Since several features relied on previous information, initial inspections of a given facility were not modelled.

Classification of critical violations

Logistic regression model was fit using `glm` on a training set consisting of the years 2012-2015 ($N_{\text{train}} = 6,799$). The model was tested on the year 2016 (January-November, $N_{\text{test}} = 2,466$). This amounted to an approximate 70/30 training/test split. A baseline model was also fit using the sole predictor of number of critical violations in the previous inspection. Since the baseline model uses the immediately previous time point as a predictor, this is an auto-regressive or AR[1] model. The AUC on test-set was approximately 0.72 for the baseline model and approximately 0.74 for the full model. This suggests that the performance is largely driven by the number of critical violations in the previous inspection, and that the remaining variables do not add much to the model.

In addition, penalized logistic regression and random forest models were also employed, but the performance did not differ substantially from the logistic regression model.



The AUC values for the models are given in the following table. In addition, the days saved were computed on the first 2-months of the test set using the method by the Chicago evaluation.

Model	AUC on test-set	Days saved
Baseline	0.716	4.337
Logistic	0.736	4.864
Logistic L1 Reg	0.747	4.987

Model	AUC on test-set	Days saved
Logistic L2 Reg	0.73	4.641
Logistic L1+L2 Reg	0.741	4.781
Random Forest	0.735	4.826

Because the facilities in the combined training and test sets were mostly restaurants, as shown in the following table, performance of the models did not change significantly from the above results when only restaurants were considered.

FacilityType	N
Restaurant	7175
Food Stand	1681
Meat Market	368
Mobile Food Units	41

The variable importance based on two different measures for the random forest model is given in the following plot.

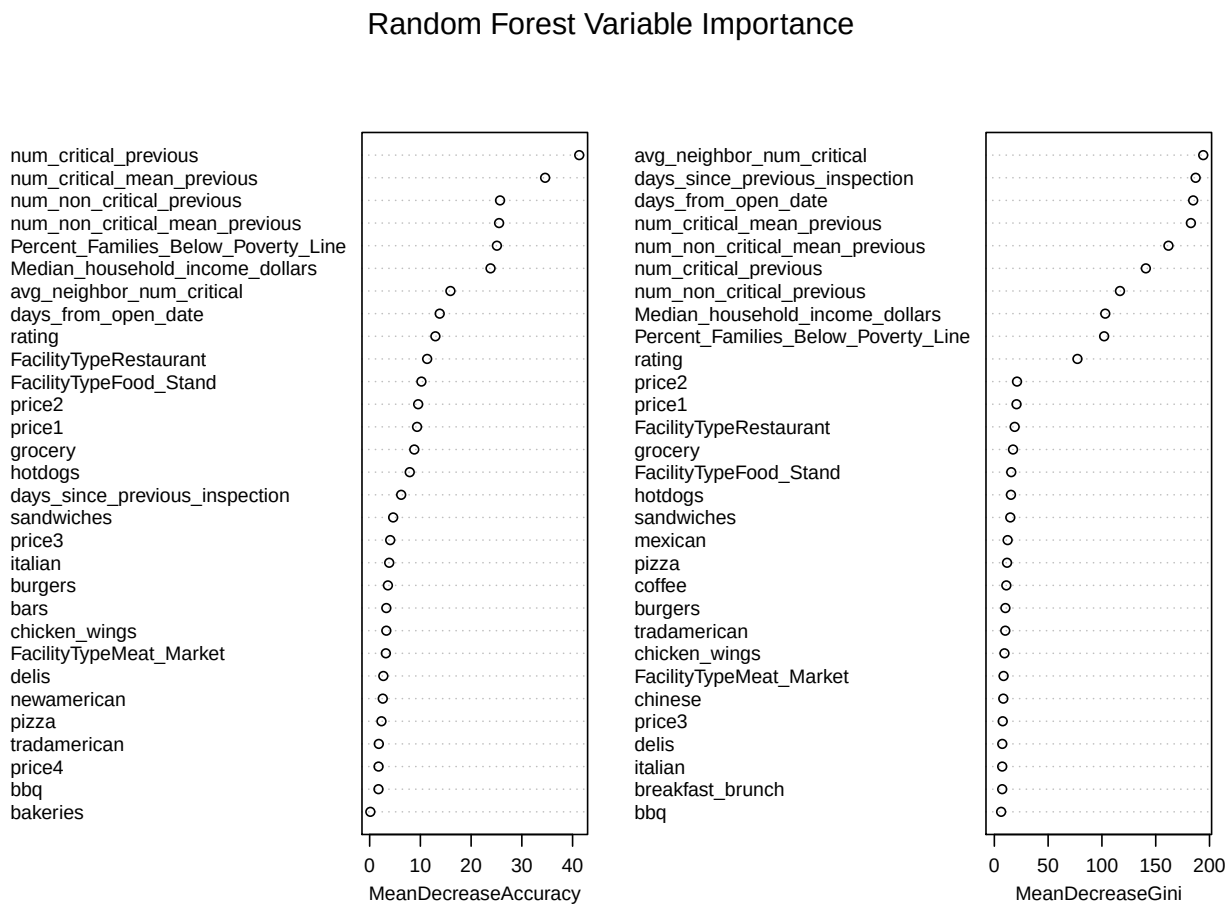


Figure 3: Variable importance for Random Forest Classification.

An important question is whether to threshold the number of critical violations. Several cutoffs were tried, such as thresholding at 2, 3, 4, or 5, and performance on the holdout test set was similar (AUC 0.73-0.77).

Modeling counts of critical violations

A Poisson regression may be more appropriate on the target variable of counts of the number of critical violations. Thus, a penalized Poisson regression model with L1 regularization was fit. Cross-validation on the training set was used to select the optimal λ that minimized cross-validation error. This resulted in the root mean squared error (RMSE) on the test-set of 2.42, which indicates that, on average, the model predictions for the number of critical violations for a given inspection deviated approximately 2.42 critical violations from the true value.

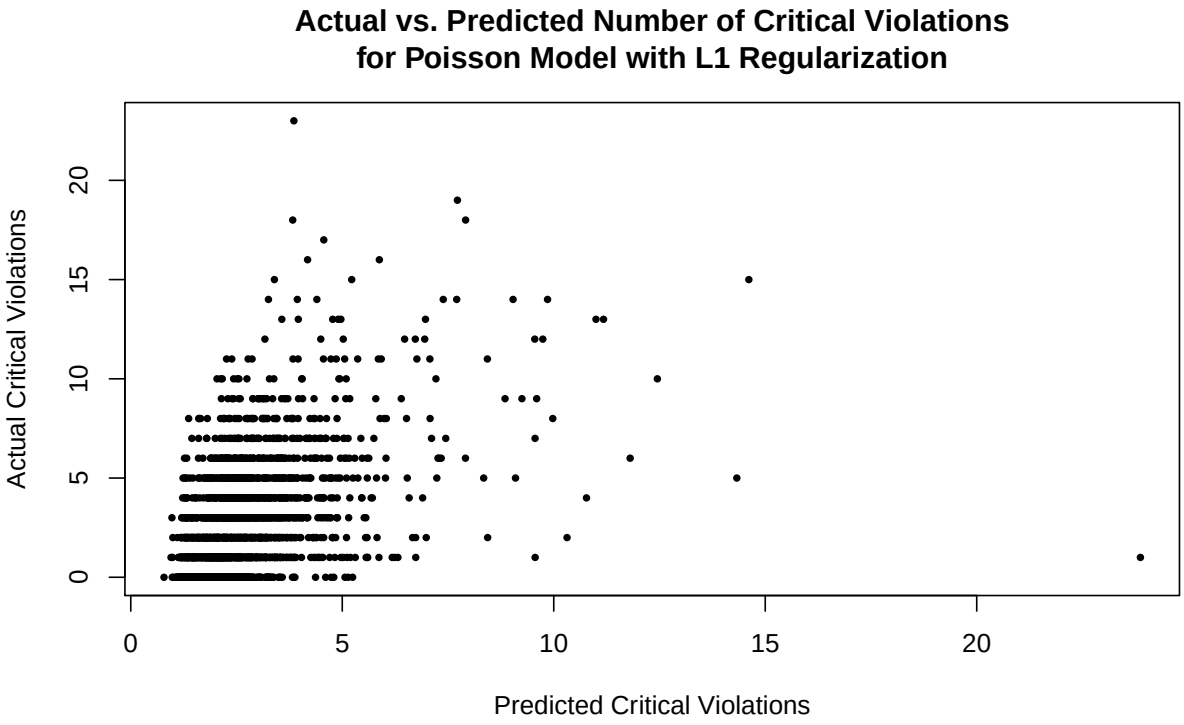


Figure 4: Actual vs. Predicted values for Poisson Model.

In addition, the predictions of counts were used as scores from which to compute days saved. The number of days saved using the Poisson model scores in the 2-month period was 7.04 days, which is substantially higher than the 4-5 days from the classification models. The information preserved in the number of critical violations, rather than a binary threshold, seemed to have been useful to realize additional gains in sorting.

Another important question is which variables are most important in the model. Variable importance for the Random Forest model was shown previously, which revealed that previous values of inspections were important in the model, with perhaps the neighboring values being important as well. The `glmnet` package offers selecting the most regularized model that is within 1 SE of the minimum cross-validation error on the training set. This is the most simple model that offers reasonable performance. The coefficients for this model (all other variable coefficients were shrunk to 0) are shown in the following table, with results consistent with the Random Forest classification. One difference is that in the Poisson model, the indicator variable for being a restaurant is more important than it was for either of the methods of measuring variable importance for the Random Forest.

Variable	Coefficient
(Intercept)	0.295
FacilityTypeRestaurant	0.214
num_critical_previous	0.080
avg_neighbor_num_critical	0.050
num_non_critical_previous	0.022
num_critical_mean_previous	0.007

Inspectors differing

Based on the model results, it is important why the previous violations are the most important predictor for the model, particularly in light of the Chicago evaluation which used inspector information. Could it be that previous violations are important because the previous inspection was performed by the same inspector as the current one, and thus gives similar ratings? To explore this question, three avenues were pursued.

First, to examine the value of the inspector variable, inspector-only models were fit. For classification, a logistic regression model on thresholded number of critical violations resulted in a test-set AUC fit of 0.737, which is better than the baseline model using only previous violations and comparable to performance on the full model. A Poisson model on the counts of critical violations led to a RMSE on the test-set of 2.28, which is slightly better than the previous Poisson model using all predictors. Thus, it seems that inspector information is quite valuable for prediction.

Second, to examine whether the previous violations could surreptiously be making use of inspector information, the percent of inspections made by the same inspector was computed. A binary indicator was created for whether the current inspection had the same inspector as the previous inspection at the same facility. The mean of this indicator variable was 63%, indicating that a majority of inspections had the same inspector as the previous inspection at the facility. This artefact of the inspection process may introduce a stability to the violations due to how a given inspector rates restaurants, which may not translate as readily if inspectors were completely randomly assigned. This comparison is important because it can reveal whether we are capturing patterns for the restaurants or whether the modeling is subtly taking advantage of inspector behavior.

Third, to simulate what "random" assignment might entail, the data were subset to sequences of consecutive inspections where inspectors differed. Unfortunately, this reduced the training set from $N_{\text{train}} = 6,799$ to $N_{\text{train\_new}} = 2,332$ and the test set from $N_{\text{test}} = 2,466$ to $N_{\text{test\_new}} = 1,074$. The ROC curve for classification results when inspectors differ is given in the following plot.

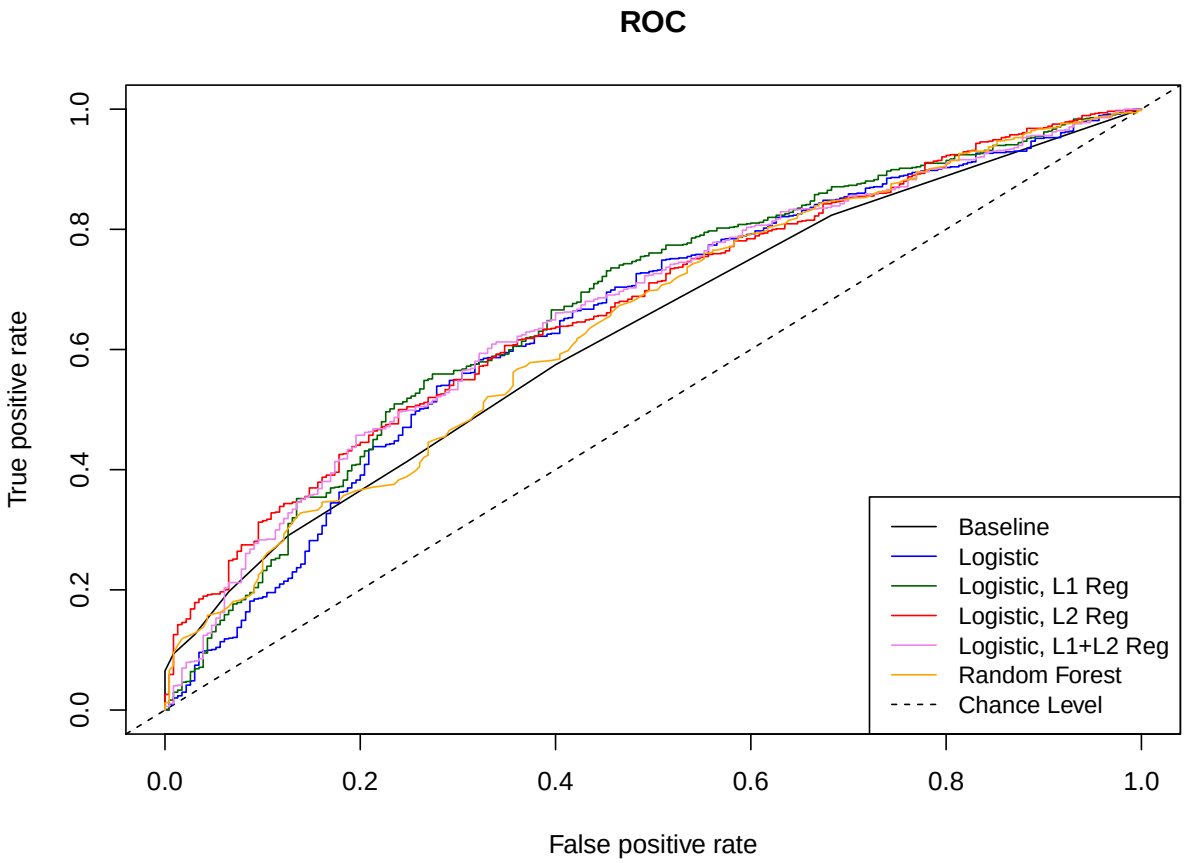


Figure 5: ROC curve for classification results when inspectors differ.

The AUC on the test-set for the various models when the inspectors differ is given in the following table. Performance decreased, although it is not clear how much this is due to fitting on a smaller dataset. A bootstrap approach to fit on the same size dataset over many bootstrap samples where inspectors do not necessarily differ could be used to see if the performance is substantially different simply due to a smaller training set, as opposed to the inspectors differing. Nevertheless, performance decreased only slightly.

Model	AUC on test-set
Baseline	0.628
Logistic	0.67
Logistic L1 Reg	0.672
Logistic L2 Reg	0.671
Logistic L1+L2 Reg	0.668
Random Forest	0.648

The same Poisson model with L1 regularization as shown previously was used to model counts on the critical violations on the subset when inspectors differ. The RMSE on the test-set was 2.69, which indicates that performance decreased, but not by very much from the 2.42 on the full model.

Importantly, it is interesting to examine the coefficients in the regularized 1 SE model when inspectors differ. In contrast to previously, where inspectors were often similar on subsequent inspections and the previous violations were important, we find that the previous violations are now less important, though they still make it into the final set of selected variables.

Variable	Coefficient
(Intercept)	0.827
FacilityTypeRestaurant	0.278
chinese	0.088
tradamerican	0.060
sushi	0.037
num_critical_previous	0.024
num_critical_mean_previous	0.023

Since (1) there is high inspector repetition found in the inspection data, (2) inspector-only models were comparable to full models without using inspector, and (3) previous violations were made less important when inspectors were forced to differ (albeit on a smaller subset of data), it seems reasonable to conclude that the number of previous violations is a proxy for inspector information, which is quite predictive on its own. Nevertheless, when inspectors are forced to differ, the performance of the models (AUC on test-set ≈ 0.67) shows that prediction is still above chance levels. In addition, we get a different picture of important variables that pertain more closely to the facility itself rather than subject to inspector behavior, such as whether the facility is a restaurant, and what type of restaurant it is (Chinese, Traditional American, or Sushi).

Conclusion

This memorandum has described the data analysis process to predict restaurant violations for Wake County, North Carolina. Prediction results were comparable to the Chicago evaluation both in terms of AUC and days saved, which was nearly 5 days out of a 2-month period. A baseline model using only previous violations was found to be satisfactory and is a recommended starting point, particularly since it only uses inspection data. Further analysis attempted to characterize to what extent the previous violations serve as a proxy for the inspector ID. Upon subsetting the data to consecutive inspections where inspectors differ, performance decreased only slightly and variables more relevant to the facility (such as whether it is a restaurant, and type of cuisine) became more important. Finally, a Poisson model was fit to the counts of violations. By preserving

the count information to order the inspections, the number of days saved increased to approximately 7 days over a 2-month period.