Does 4 Stars mean a Clean Kitchen? Predicting Restaurant Health Inspection Results using Yelp Data

Chris Le and Graham MacDonald

Executive Summary

This study attempts to predict restaurant inspection violations using data available on the review site Yelp. Predicting inspections could help inform the public and help local health departments become more effective by better targeting scarce inspection resources. Both actions could reduce hospital trips, improve health outcomes, and reduce health expenditures. We find that our best predictive model improves accuracy by 10% over baseline accuracy, a relatively small improvement.

More research needs to be done to improve the model's accuracy to make the improvements in targeting worth the effort for local health departments. Given that there are likely significant costs of implementation in gathering the data and implementing the model on a regular basis, the benefit to citizens' health and the cost savings to the department should be considered before implementing this model.

Introduction

Local Health Inspections

Local, municipal governments – usually city or county agencies – perform restaurant health inspections in the United States. After completing the inspections, the local agencies impose fines, follow-up to ensure violations are corrected, and sometimes shut down businesses for non-compliance. Some particularly entrepreneurial cities, like New York, also require that restaurants post grades from their inspections in a visible space on their storefront, and others, like San Francisco, submit their data to Yelp to be posted online next to customer ratings.

Most municipalities, however, never see the inspection scores assigned by the local heath department. In some major cities, the department is not proactive about getting the information out, like San Francisco or New York, but has the data available through an open data portal, for anyone to download and distribute. Most agencies, however, don't post the data at all, or when they do, it is in a format that is very difficult for the average user to access, read through, and interpret.

Can Everyone Have Access to Good Data?

Many local governments don't proactively post good data or ally with Yelp or restaurant owners because they face small, and in many cases, declining budgets. These organizations can barely sustain current operations, never mind launching efforts to create clean, public-friendly interfaces or even commit to regularly posting updated data in an accessible format for developers. Although it is possible that as the economy recovers, local agencies could find budget money to make open data a priority, it is unlikely that areas outside of major cities would have the wherewithal – or community pressure – to push this agenda.

Yelp bet on such a movement with its adoption of the LIVES standards – data format standards for local agencies to easily upload data to Yelp – in 2012, and since adoption a number of major cities have joined the initiative. However, three years after the initiative was announced, many places in large, dense metropolitan areas, including creative, progressive municipalities like the City of Berkeley, still haven't adopted the open standard.

So, What Can We Do?
So how can all Americans easily access information about their health inspection scores, even if they don't live in a major city? Yelp contains an incredible wealth of information about establishments, including but not limited to user reviews, restaurant hours, prices, restaurant type, location, and even small details such as whether the place has a TV or the "ambience".

Yelp data isn't limited by jurisdictional boundaries, is easily available, and is widely adopted. What if all the rich information from Yelp and its users could accurately predict restaurant violations? At low cost to local governments, all Americans could benefit from accurate, easily accessible health code information, which can help people reduce medical costs and avoid unpleasant meals out. As Code For America notes, when the City of Los Angeles forced restaurants to display inspection grades, the city saw a corresponding "13% decrease in hospitalizations due to food-borne illness."[1]

Predicting violations could also help health departments better target their limited resources by sending inspectors to restaurants with higher predicted violations scores. When restaurants with poor standards notice that the health department is more effective at targeting, they could change their behavior or shut down, which would lead to higher standards and healthier customers.

This paper attempts to determine whether Yelp data can accurately predict restaurant critical health violations by examining data collected from the nation's capital.

About the Data
Sources
Data for this analysis comes from three sources: data scraped from the DC Department of Health Online Food Facilities Inspections webpage[2], cleaned and geocoded by Graham MacDonald's website Inspectionmapper[3], data scraped from Yelp.com, and 2009-2013 block group level data from the American Community Survey.

As part of the Inspectionmapper website, restaurant inspections are manually matched to Yelp Business IDs by Graham when he updates the data monthly, so there is no need to do complex entity matching across datasets. Data are geocoded monthly using the open

---

[1] http://www.codeforamerica.org/our-work/data-formats/LIVES/
[2] http://washington.dc.gegov.com/webadmin/dhd_431/web/
[3] http://grahamimac.com/inspectionmapper/

source Nominatim API[4], which is based on publicly available OpenStreetMap data. Block group data from the Census and the corresponding shapefiles were downloaded from NHGIS[5], and restaurants were assigned to block groups using a point in polygon operation.

Overview

All data is for restaurants located in DC.Yelp.com restaurant level data represent data for 2015, while review data is based on all Yelp reviews for that business since it first opened.

Each observation in the analysis represents a unique restaurant in DC with at least one routine or complaint inspection between January 2014 and March 2015. Other inspections, such as preoperational, follow-up, and renewal inspections, were not included because restaurants can easily anticipate these and affect their results. The variable we want to predict is the average number of critical violations per inspection for a given restaurant.

Why not predict individual inspection violations instead of the average of violations across inspections? Inspections of any kind are susceptible to the bias of the inspector, and different inspectors at different times and different days or years might give the same restaurant in the same condition a different number of violations, depending on how strict they are or how many mistakes employees happened to make the day of the inspection. By taking the average of the number of critical violations, we get a better idea for the underlying food safety of the restaurant over a number of different inspections, as opposed to the day and time specific factors. And by using data from just the past year, we eliminate the effects of inspectors giving greater or fewer inspections over time.

The data are separated into a training set, representing a randomly selected 70% of the data, and a test set, representing the remaining 30%.

The data used for the analysis can be separated into four distinct categories – inspection data, location data, Yelp restaurant data, and Yelp review data – and are summarized in the table below. The raw inspection dataset contains 9,980 observations and 16 features, the raw block-group level data contains 450 observations and 5 variables, the Yelp restaurant data contains 1,421 observations and 29 variables, and the Yelp review data contains 231,952 observations and 9 variables. The combined training set with feature engineering contains 584 observations and 130 variables, while the combined test set contains 249 observations and 130 variables.

---

[4] http://wiki.openstreetmap.org/wiki/Nominatim

[5] Minnesota Population Center. *National Historical Geographic Information System: Version 2.0*. Minneapolis, MN: University of Minnesota 2011. http://www.nhgis.org

Variables Used in the Analysis

Inspection Data
- Average number of critical violations
- Average of pre-2014 critical violations
- Worst Violations (> 2)

Location Data
- Block group median income
- Block group share of population ages 15-29
- Block group share of population ages 30-44
- Block group population per square meter x 100
- Zip code

Yelp Restaurant Data
- Price (1 to 4 $s)
- Number of reviews
- Number of stars
- Number of stars weighted by reviewer # of reviews
- Accepts credit cards
- Serves beer and wine only
- Has a full bar
- Ambience – Intimate, touristy, upscale, divey, classy, trendy, hipster, casual
- Is the attire casual
- Has bike parking
- Caters
- Delivers
- Good For – Lunch, dinner, breakfast, dessert, late night, groups, kids
- Has a TV
- Noise level – Quiet, loud, very loud
- Has outdoor seating

Yelp Restaurant Data (continued)
- Parking – Private lot, garage, street, valet
- Has take-out
- Takes reservations
- Has waiter service
- Is wheelchair accessible
- Has wi-fi
- Type – Bakeries, French, breakfast & brunch, Indian, seafood, Latin, burgers, bars, Thai, delis, Italian, American traditional, Mexican, pizza, American new, fast, Chinese, sandwiches
- Open 24 hours
- Open continuously from open to close
- Open time – Early morning, morning, evening
- Close time – Evening, night, late night, very late night

Yelp Review Data
- Unigram of all “dirty” words – Dirty, foul, stain, filthy, unclean, grimy, grubby, messy, shabby, dilapidated, etc.
- Unigram of all “sick” words – Sick, ill, puke, vomit, nausea, barf, queasy, etc.
- Unigram of all “clean” words – Clean, fresh, neat, tidy, orderly, sparkling, spotless, etc.
- Share of reviews with less than 3 stars
- Highly related unigrams (single words), bigrams, 3-, 4-, and 5-grams

Feature Engineering
In order to prepare the dataset for analysis, we created a number of new variables from the original data and recoded certain missing values.

Violations

- **Critical violations prior to 2014** is the restaurant's average number of critical violations before 2014
- The **worst violations** are defined as restaurants with an average number of critical violations above 2

Location

- The **zip code** variable is a set of binary variables for each zip code that takes the value of 1 if the restaurant is located in that zip code and 0 otherwise

Yelp Restaurant Data

- **Number of stars weighted by number of reviews** is the weighted average of star ratings weighted by each reviewer's total number of reviews on the site. This theoretically gives more weight to more experience
- **Ambience, good for, noise level, parking, type, open time,** and **close time** are binary variables for each of the items in the given category, where the variable takes a value of 1 if the restaurant is in a given category, and 0 otherwise
- **Open 24 hours** is a binary variable that takes a value of 1 if the restaurant is open 24 hours and 0 otherwise
- **Open continuously** is a binary variables that takes a value of 1 if the restaurant does not close and re-open at some point during the day and 0 otherwise
- **Open time** is a set of binary variables with time periods defined as follows – **early morning** is an open time before 9 a.m. – **morning** is an open time from 9 a.m. to before 1 p.m. – **evening** is an open time after 4 p.m. to 7 p.m.
- **Close time** is a set of binary variables with time periods defined as follows – **evening** is a close time after 4 p.m. to 7 p.m. – **night** is a close time after 7 p.m. and before midnight – **late night** is a close time from midnight to 2 a.m. – **very late night** is a close time after 2 a.m. to 6 a.m.

Yelp Review Data

- **Uni-, Bi-, 3-, 4-,** and **5-grams** calculate the relative frequency of a word in a restaurant's reviews compared to the frequency of that same word or group of words in all restaurant reviews overall
- **"Dirty" words** represents unigram of a group of words related to "dirty" from the Corpus of Global Web-Based English[6]
- **"Sick" words** represents unigram of a group of words related to "sick" from the Corpus of Global Web-Based English[7]
- **"Clean" words** represents unigram of a group of words related to "clean" from the Corpus of Global Web-Based English[8]

---

[6] http://corpus.byu.edu/glowbe/
[7] http://corpus.byu.edu/glowbe/

- **Share of reviews with less than 3 stars (a.k.a. "bad")** is a variable that represents the share of reviews for a restaurant that receive less than 3 stars
- **Highly related grams** represent words and phrases between 2 and 5 words long that were the top 3 most important features when used to predict critical violations, the worst violations, and "gross" violations in the training set in each category of grams (i.e., there are 3 for unigrams for critical violations, 3 for bi-grams, 3 for 3-grams, … 3 for unigrams for the worst violations, 3 for bi-grams for the worst violations, etc.)

Missing Values

- **All variables'** missing values are replaced by the column mean, with a few exceptions
- **Highly related grams'** missing values are replaced with a zero for missing values, and –infinity values are recoded to -1000

Summary Statistics

See Appendix A for complete tables.

Introduction to Approaches

We attempted two predictive models. One model attempts to predict the average number of critical violations – a regression – while the other attempts to predict restaurants with more than 2 critical violations – a classification.

Each team member worked on a few approaches to feature engineering and analyzed the data using a few different models.

Graham

Graham worked on different methods of analyzing words, bad reviews, and star rating. He also tuned a number of different models to predict the continuous variable critical violations, and tuned a few of the same approaches on the binary variable.

Chris

Chris worked on models aimed at leveraging the review text to find words with some predictive power. After extensive feature engineering, he ran a number of models using both regression as well as classification.

Graham's Section

Number of Critical Violations [Graham]

To predict the number of critical violations, I used a number of different regression models – a linear regression, a ridge regression, a lasso regression, a Bayesian ridge regression, a SVM regression – with RBF kernel – a random forest, AdaBoost on a random forest, and many AdaBoost models on a random forest using k-means bagging. The error measurement for the regression analysis was root mean squared error, the

[8] http://corpus.byu.edu/glowbe/

square root of the mean of squared errors in the regression analysis. The lower the root mean squared error (RMSE), the better the prediction.

For the ridge, lasso, and Bayesian ridge regressions, I tuned the alpha parameter along a number of values from 0.00001 to 500 to find the parameter than minimized RMSE. I did the same – with a similar set of values – for the SVM regression on the C parameter. For the random forest, I tuned a number of different values for the maximum number of leafs and the maximum depth to find the parameters that minimized RMSE. I used 100 trees for a more stable model, and re-ran the most accurate model using 1000 trees. For the AdaBoost model, I tuned the number of estimators and the learning rate on a standard random forest regressor. I ran the most accurate model on a standard random forest regressor with 100 trees, as opposed to 10. All models were tuned using 10-fold cross-validation on the training set, and the most accurate model was tested on the test set.

Models, Tuning Parameters, Accuracy, and Improvement over Baseline [Graham]

Model	Best Tuning Parameters	MSE	RMSE	Improvement Over Baseline
SVM	C=4	7.11	2.67	-1%
Baseline	-	7.01	2.65	0%
Linear	-	6.70	2.59	2%
Bayesian Ridge	alpha_1=0.00001, lambda_1=1	6.68	2.58	2%
Lasso	alpha=0.04	6.31	2.51	5%
Ridge	alpha=60	6.16	2.48	6%
Random Forest	min_samples_leaf=4, max_depth=10	6.01	2.45	7%
AdaBoost (RF)	n_estimators=100, learning_rate=0.5	5.71	2.39	10%
AdaBoost (RF) w/ K-means Bagging	k=12	5.63	2.37	10%

The best model was using 20 differently tuned AdaBoost models on a random forest regressor, with predictions bagged using k-means bagging, which produced a RMSE of 2.37, a reduction of 10% over the baseline RMSE. K-means bagging is the process of using 10-fold cross validation to create out of sample predictions for very similar, highly accurate models with slightly different tuning parameters, and running k-means on these predictions to group them into similar clusters. The entire range of possible k values, or number of groups, is tested against the RMSE to find the number of groups that minimizes RMSE. Predictions within groups are averaged together, and then these in-group prediction averages are averaged together to get the final prediction.

K-means bagging essentially gives lower weight to similar predictions, and higher weights to different predictions – because the predictions are all highly accurate, this should produce a more accurate prediction overall. And k-means bagging does produce a modest increase in accuracy, reducing RMSE from 2.39 to 2.37 over the single best AdaBoost model.

The RMSE here is still very high, more than 2 critical violations – compared to a mean of just over 3 in the training set – indicating that it is very difficult to predict with much accuracy the number of critical violations a restaurant might have. Confirming this conclusion, there was no single indicator that was much more important than the others – though many of the n-grams, the number of reviews, the share of bad reviews, and the number of previous critical violations were the most important features, there were no features with a feature importance above 0.07.

Single Best AdaBoost Model (RF), Important Features (>= 0.02) [Graham]

Feature	Feature Importance
but if you	0.067
bad	0.050
PrevCriticalViolations	0.044
reviews	0.042
clean	0.041
the best ive ever had	0.040
Block_Ages_30_44	0.040
Block_100Pop_Sq_Meter	0.033
the only thing that was	0.030
sick	0.030
Block_Ages_15_29	0.029
we ordered were	0.026
dirty	0.026
Block_Med_Inc	0.023
price	0.021

Greater than 2 Critical Violations [Graham]

To predict restaurants with more than 2 critical violations, I used a number of different classification models that I had found most powerful in previous classification analysis – a logistic regression, a SVM – with RBF kernel – a random forest, and AdaBoost on a random forest. The error measurement was the percent of observations correctly predicted.

For the SVM classifier, I tuned a number of different values for the C parameter from 0.00001 to 5 to find the model with the highest accuracy. For the random forest, I tuned different values for the maximum number of leafs and the maximum depth of the tree to find the model with the highest accuracy. I used 100 trees for a more stable model, and re-ran the most accurate model using 1000 trees. For the AdaBoost model, I tuned the number of estimators and the learning rate on a standard random forest classifier. I ran the most accurate model on a standard random forest classifier with 100 trees, as opposed to 10. All models were tuned using 10-fold cross-validation on the training set, and the most accurate model was tested on the test set.

Models, Tuning Parameters, Accuracy, and Improvement over Baseline [Graham]

Model	Best Tuning Parameters	Accuracy	Improvement over Baseline
Logistic	-	53.0%	-15%
Baseline	-	62.7%	0%
SVM	C=0.00001	63.1%	1%
Random Forest	min_samples_leaf=4, max_depth=5	64.7%	3%
AdaBoost (RF)	n_estimators=100, learning_rate=0.7	65.9%	5%

The best model was the AdaBoost on a random forest classifier, with an accuracy of 66%, a 5% improvement over the baseline accuracy. While the accuracy is a little higher than baseline, but not so much higher that it is clear it should be used in targeting city restaurants for inspection. Similar to the continuous prediction, there was no single indicator that was much more important than the others – though some of the grouped unigrams, the number of reviews, block specific census data, and the number of previous critical violations were the most important features, there were no features with a feature importance above 0.06.

AdaBoost Model (RF), Important Features (>= 0.02) [Graham]

Feature	Feature Importance
reviews	0.053
Block_Ages_30_44	0.045
PrevCriticalViolations	0.042
Block_100Pop_Sq_Meter	0.039
Block_Med_Inc	0.039
clean	0.039
bad	0.034
Block_Ages_15_29	0.032
sick	0.032
dirty	0.028
stars	0.020

Chris's Section

Preprocessing [Chris]

The bulk of my work came from trying to establish good features that would hopefully yield some sort of correlation to the prediction variable. Most of these features involved the reviews themselves and not much about the business.

Word Features (without feature extraction from sklearn) [Chris]

I created the word features manually to try to learn the process and possibly gain some intuition and flexibility with the extraction. I first concatenated all of the reviews correlating to each respective business. Then I went through each restaurant and added each unique word (punctuation removed) to a dictionary for that restaurant as well as a dictionary that held all unique words. After doing this for all of the restaurants, I was able to create unique word features for each restaurant that had the ratio of how many times a certain word was mentioned for a restaurant with respect to all of the restaurants. This would tell us if there was an abnormal amount of specific words (for example “dirty”). In order to keep the number of features reasonable, I took the top 3000 words and then removed all stop words such as “to”, “for”, “the”. The following is an example of ten restaurants and three words.

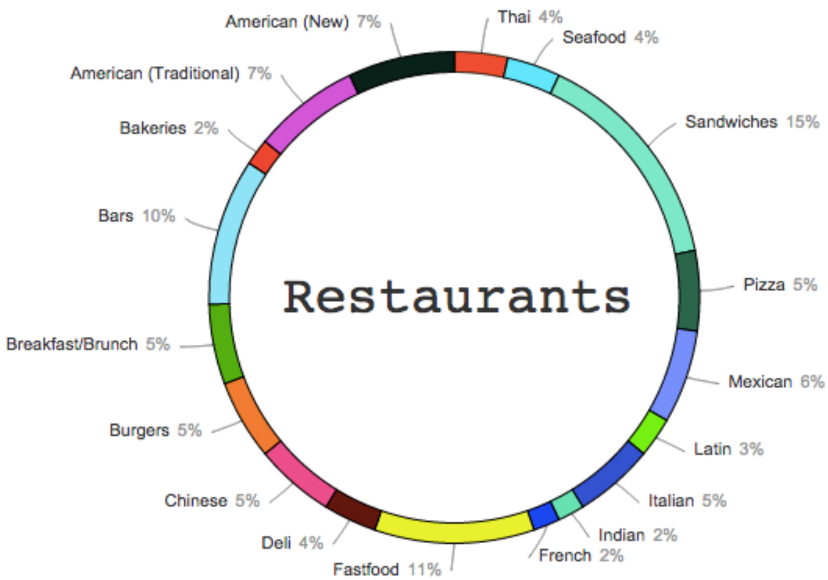
dirty	sick	clean
0.010752688	0	0.161290323
0.058823529	0	0.088235294
0	0	0.1
0	0	0.068965517
0.063492063	0.015873016	0.111111111
0.058823529	0.019607843	0.078431373
0.0625	0	0.073863636
0.008849558	0	0.088495575
0.015151515	0.007575758	0.090909091
0.027027027	0.027027027	0.135135135

Negative Trend Features [Chris]
I created a new feature called “negative trend” that tracked the month before the inspection to see if there was any decrease in stars given. I also created a new feature that looked specifically at words synonymous with dirty, unclean, bugs, hair, etc. that could possibly yield a health violation. I tracked their occurrences within that one-month period. I wanted to see if an abnormal occurrence of the word was mentioned. If this ratio were higher than the ratio of its frequency throughout all of the reviews it would display 1 and conversely 0.

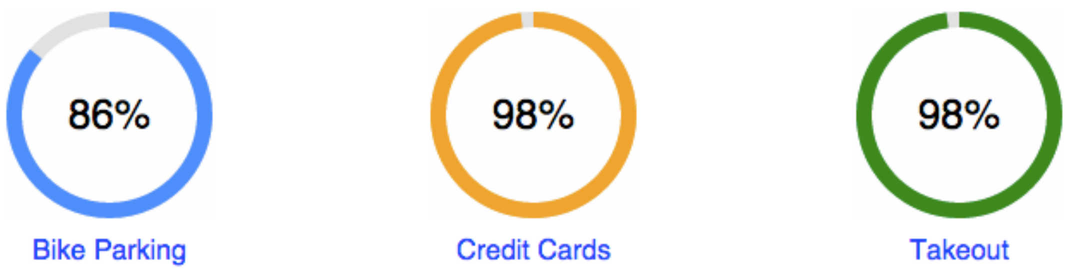
Data Visualization [Chris]
Originally, I had planned on creating an html slide deck using some templates I found online but then I realized that my javascript, html, and css skills are not too great so it took me almost three hours to do one slide. I then just took screenshots of the charts I made by naively tweaking some of the D3 examples and included them into a PowerPoint (this removed animations and interactivity).

Pie chart including all of the unique types of restaurants:

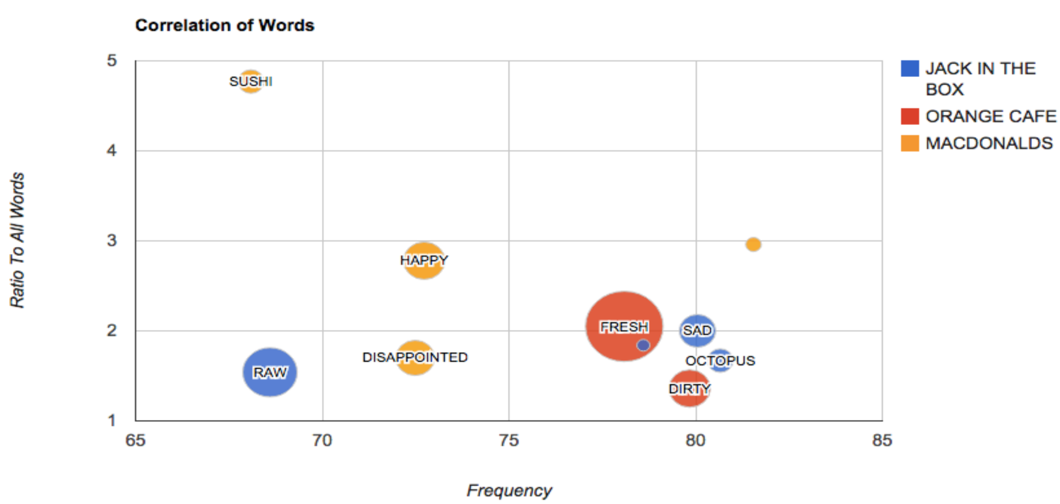
[Chris]



Progress pies include three features in our data:



Bubble map with samples of words and restaurants (I could not quite figure out how to properly change the axes):



Models [Chris]
Using the dataset that I created and then concatenating it with some of Graham's features, I began to try to run some basic models to see if there was any real improvement over baseline. The models includes:

- SVM with RBF kernel
- Decision Tree Regression
- Extra Trees Regression
- Extra Trees Classifier
- Gaussian Naïve Bayes
- Multinomial Naïve Bayes
- Random Forest
- Adaboost with Random Forest, Extra Trees, and SVMs

The best performance from all of these models came from the Adaboost with the Random Forest without any tuning. Since Graham already had a superior model, I proceeded to try to create more features while Graham tuned this optimal model for what we currently had.

Error Testing [Chris]
In order to test, I used k-fold cross validation with k = 10. This allowed me to hold out 10% of the data to validate against the other 90%. By taking the mean of the results, I was able to get a pretty accurate prediction of how well my model would perform. Ultimately, even without much tuning I found that all of my models made very insignificant improvements over baseline. I also ran a confusion matrix on my models to see if I was overfitting. It did not show signs of overfitting for most of the models and really just had low correlation.

Conclusion
We found that the best model predicting the average number of critical violations in restaurants in Washington, DC in 2014 and 2015 using information available on Yelp improved prediction accuracy by 10% over baseline, from an error of 2.65 to 2.37. We also found that the best model predicting the number of restaurants with more than 2 critical violations, on average, during the same period improved prediction accuracy by 5%, from 63% to 66%.

We collected data from four different sources and engineered a number of different theoretically defensible features – for a total of 130 different variables overall – and none stood out as being particularly important in predicting critical violations. We also tried a number of different modeling approaches, including a SVM, linear regression, Bayesian ridge regression, lasso regression, ridge regression, logistic regression, random forest, AdaBoost on a random forest, and AdaBoosts on a random forest bagged with k-means. The best continuous model was a k-means bagging of 20 different AdaBoost models with k=12 on a random forest, which slightly outperformed the best tuned AdaBoost model on a random forest. The best binary model was the best-tuned AdaBoost model on a random forest.

Real-World Implications

Given the small improvement in accuracy of the two models, and the large remaining error, we would not recommend that health departments rely on this predictive model to improve targeting in the field. It is possible that this small amount of improvement might be enough to help better target restaurants for inspection, but given the large error (2.37) compared to the mean number of violations (3.3), it is unlikely that the current predictions will prove useful. More research needs to be done to improve the accuracy of the model before it becomes clear that the cost of staff time to implement the model is worth the improvement in health outcomes realized from using it.

Appendix A
The following represents summary statistics calculated from the training set before missing values were replaced with column means.

Variable	Mean	Median	Standard Deviation
CriticalViolations	3.30	3.00	2.49
Previous Critical Violations	2.47	2.33	1.44
Worst	41%	0%	49%
Block_Med_Inc	$96,934	$91,576	$43,846
Block_Ages_15_29	16%	11%	17%
Block_Ages_30_44	27%	28%	10%
Block_100Pop_Sq_Meter	0.58	0.43	0.54
gross	7%	0%	26%
price	1.7	2.0	0.7
reviews	187	104	271
stars	3.3	3.5	0.6
timeContinuous	94%	100%	24%
octopus	-894	-1,000	308
edible	-801	-1,000	399
events	-844	-1,000	363
and left	-769	-1,000	422
menu the	-755	-1,000	430
bill came	-890	-1,000	313
we ordered were	-890	-1,000	313
but if you	-724	-1,000	447
for a glass	-902	-1,000	297
service was a little	-873	-1,000	333
a glass of wine	-870	-1,000	337
and the service was	-705	-1,000	456
the best ive ever had	-841	-1,000	366
the only thing that was	-913	-1,000	283
my husband and i were	-916	-1,000	278
food ive ever had	-918	-1,000	275
the food was not	-866	-1,000	341
in the near future	-911	-1,000	285
i decided to give it	-918	-1,000	275
would definitely come back here	-914	-1,000	280
a lot to be desired	-895	-1,000	306
dirty	-433	-1	496
sick	-329	0	470
clean	-161	0	368

Variable	Mean	Median	Standard Deviation
bad	28%	26%	22%
PrevCriticalViolations	2.5	2.5	1.4
WasComplaint	24%	0%	58%
CreditCards	98%	100%	13%
IsBeerWine	7%	0%	26%
IsFullBar	51%	100%	50%
Ambience_Intimate	4%	0%	19%
Ambience_Touristy	1%	0%	9%
Ambience_Hipster	2%	0%	14%
Ambience_Divey	6%	0%	24%
Ambience_Classy	7%	0%	26%
Ambience_Trendy	8%	0%	28%
Ambience_Upscale	2%	0%	12%
Ambience_Casual	70%	100%	46%
Attire_Casual	94%	100%	24%
Bike_Parking	86%	100%	35%
Catering	49%	100%	50%
IsDelivery	21%	0%	41%
GoodFor_Lunch	48%	100%	50%
GoodFor_Dinner	51%	100%	50%
GoodFor_Breakfast	8%	0%	28%
GoodFor_Dessert	1%	0%	8%
GoodFor_LateNight	8%	0%	27%
GoodFor_Groups	81%	100%	39%
GoodFor_Kids	71%	100%	46%
HasTV	43%	0%	50%
Noise_Quiet	12%	0%	33%
Noise_Loud	9%	0%	28%
Noise_VeryLoud	3%	0%	17%
Outdoor_Seating	48%	100%	50%

Variable	Mean	Median	Standard Deviation
Parking_PrivateLot	8%	0%	27%
Parking_Garage	10%	0%	30%
Parking_Street	91%	100%	29%
Parking_Valet	8%	0%	27%
IsTakeOut	91%	100%	29%
TakesReservations	41%	0%	49%
WaiterService	58%	100%	49%
WheelchairAccessible	78%	100%	41%
WiFi	34%	0%	47%
RestType_Bakeries	2%	0%	12%
RestType_French	2%	0%	14%
RestType_Breakfast&Brunch	6%	0%	24%
RestType_Indian	2%	0%	14%
RestType_Seafood	4%	0%	19%
RestType_Latin	3%	0%	16%
RestType_Burgers	6%	0%	24%
RestType_Bars	11%	0%	31%
RestType_Thai	4%	0%	19%
RestType_Delis	4%	0%	20%
RestType_Italian	6%	0%	23%
RestType_American (Traditional)	8%	0%	27%
RestType_Mexican	7%	0%	25%
RestType_Pizza	6%	0%	23%
RestType_American (New)	8%	0%	28%
RestType_Fast	12%	0%	32%
RestType_Chinese	6%	0%	23%
RestType_Sandwiches	17%	0%	38%
Open24Hours	1%	0%	12%
Open_Early	18%	0%	38%
Open_Morning	71%	100%	46%
Open_Evening	8%	0%	27%
Close_Evening	6%	0%	24%
Close_Night	57%	100%	50%
Close_LateNight	20%	0%	40%
Close_VeryLateNight	13%	0%	34%

Variable	Mean	Median	Standard Deviation
20016	4%	0%	19%
20001	13%	0%	33%
20002	9%	0%	28%
20003	8%	0%	27%
20008	5%	0%	21%
20036	9%	0%	28%
20009	10%	0%	30%
20015	2%	0%	12%
20019	0%	0%	6%
20020	1%	0%	10%
20004	6%	0%	24%
20007	6%	0%	24%
20018	2%	0%	12%
20024	3%	0%	17%
20005	7%	0%	26%
20006	5%	0%	23%
20010	2%	0%	14%
20037	3%	0%	17%
20017	1%	0%	11%
20011	2%	0%	14%
20565	0%	0%	4%
20012	1%	0%	10%
20045	0%	0%	4%
20057	0%	0%	6%
20032	1%	0%	7%
20052	0%	0%	0%
20566	0%	0%	4%
20023	0%	0%	0%
20050	0%	0%	4%