PREDICTING HEALTH CODE VIOLATIONS IN BOSTON RESTAURANTS

Springboard Capstone Project
by Roy Wright, August 2016

1. PROBLEM STATEMENT

CLIENT AND PROBLEM

The City of Boston, like many others, conducts health inspections of food service establishments in a largely random pattern of visits. With a model for predicting which of all the potential inspections will result in violations, the inspections can be carried out in a targeted way. This could make more efficient use of inspectors' time and, more importantly, increase the chances of detecting violations.

Our aim is to find restaurants that are likely to fail an unannounced health inspection — that is, we will develop the capability of predicting whether a health code violation would be found at a given food service establishment, if it were chosen to be inspected, without prior notice, at the moment that the predictive model is used. The main factors to be used in predicting an inspection outcome are the urban conditions near the restaurant in the weeks leading up to the inspection.

DATA USED

One dataset that we make very limited use of contains records of health inspections throughout Boston, from 2006 to present, obtained from data.cityofboston.gov. Each inspection record includes many pieces of information, but those that are most relevant to our purposes are the following:

- Food service establishment's name
- Type of establishment
    - category "FS" stands for "Eating & Drinking"
    - category "FT" stands for "Eating & Drinking w/ Take Out"
    - category "MFW" stands for "Mobile Food Walk On"
    - category "RF" stands for "Retail Food"
- The date of the inspection
- Description of each violation found, or of each previous violation that has been corrected, along with its severity, rated as * or ** or *** (one star, two stars, or three stars)
- The result of the inspection
    - "HE_Pass" when an establishment passes an inspection with no violations (either its first yearly inspection or a re-inspection a few weeks after failure)
    - "HE_Fail" when an establishment's first inspection of the year is unsatisfactory – which occurs if any violation(s) is/are found, even as little as a single one-star issue
    - "HE_FailExt" when an establishment is re-inspected within a few weeks after a failure, and fails again
    - "HE_Filed" for the first inspection of an opening establishment
    - other result codes – for example, temporary business closure because of an emergency – are much less common, and do not pertain to the type of inspection studied here
- Written comments
- Business address

- Latitude/longitude coordinates – unfortunately, this information is not always provided as part of the inspection

Another dataset that is more useful to our purposes comes from Yelp.com, as part of the 2015 "Keeping it Fresh: Predict Restaurant Inspections" contest, hosted by DrivenData.org. At the heart of this dataset is a list of Boston restaurant inspection results. Each row gives the inspection date, a "restaurant ID" for the establishment inspected, and tallies of the violations found at each severity level (*, **, and ***). Note that an inspection is passed if and only if there are *zero* violations found at *each* of these severity levels. The latitude and longitude coordinates of each establishment are also given.

Now, we have access to two separate datasets that provide records of past health inspections. To compare these datasets, let's take a look at the contents of each one for a single day. Below, we display a small portion part of the contents of the City of Boston's records for August 12, 2014.

name	**category**	**result**	**level**	**description**	**comments**	**latitude**	**longitude**
DUNKIN DONUTS(FRANKLIN)	FT	HE_Pass	NaN	NaN	NaN	42.356510	-71.053320
KANTIN	FT	HE_Fail	***	PIC Performing Duties	The time as a public health control logs are n...	42.352411	-71.125329
KANTIN	FT	HE_Fail	*	Installed and Maintained	The cold water at the back handwash sink is no...	42.352411	-71.125329
KANTIN	FT	HE_Fail	*	Non-Food Contact Surfaces	There is duct tape on the handle of the rice c...	42.352411	-71.125329
KANTIN	FT	HE_Fail	*	Premises Maintained	There is excess clutter in the upstairs storag...	42.352411	-71.125329
Samurai Kuang Eatery	FT	HE_Pass	***	Cold Holding	Sushi grade salmon 51F White fish 50F / Provi...	42.355795	-71.058451
Samurai Kuang Eatery	FT	HE_Pass	*	Equipment Thermometers	Dish machine gauge is broken / Repair.	42.355795	-71.058451
Samurai Kuang Eatery	FT	HE_Pass	*	Improper Maintenance of Floors	Floors under cookline around handsink heavily...	42.355795	-71.058451
Samurai Kuang Eatery	FT	HE_Pass	*	Non-Food Contact Surfaces	Back door opened without screen / Provide scr...	42.355795	-71.058451
Samurai Kuang Eatery	FT	HE_Pass	*	Improper Maintenance of Walls/Ceilings	Hood vents with visible grease build up / Clea...	42.355795	-71.058451

name	category	result	level	description	comments	latitude	longitude
Shaw's Supermarket No. 586	RF	HE_Fail	*	Food Contact Surfaces Design	Sponge being used at the 3 bay sink in the pro...	42.271930	-71.069700
Shaw's Supermarket No. 586	RF	HE_Fail	*	Non-Food Contact Surfaces Clean	Interior of the chicken freezer near the rotis...	42.271930	-71.069700
Shaw's Supermarket No. 586	RF	HE_Fail	*	Improper Maintenance of Floors	Floor under the storage cabinet near the rotti...	42.271930	-71.069700
Shaw's Supermarket No. 586	RF	HE_Fail	*	Improper Maintenance of Walls/Ceilings	Portion of the wall in the meat walk-in cooler...	42.271930	-71.069700
Shaw's Supermarket No. 586	RF	HE_Fail	*	Installed and Maintained	Pipe under the hand sink in the meat preparati...	42.271930	-71.069700
Shaw's Supermarket No. 586	RF	HE_Fail	*	Non-Food Contact Surfaces	Salad bar unit operating at around 50F. PIC (B...	42.271930	-71.069700
Shaw's Supermarket No. 586	RF	HE_Fail	***	Cold Holding	All foods inside of the salad bar withg temper...	42.271930	-71.069700

By contrast, here are part of the contents of Yelp's dataset for August 12, 2014:

date	restaurant_id	violations			name	latitude	longitude
		*	**	***			
2014-08-12	lnORdd3N	0	0	0	Dunkin' Donuts	42.356527	-71.053353
2014-08-12	njoZ1D3r	3	0	1	Kantin	42.352744	-71.125447
2014-08-12	B1oX4boV	4	0	1	Samurai Kuang Eatery	42.355741	-71.058335

At a glance, we can see that the inspection records maintained by the City of Boston are more detailed than those provided by Yelp. Note that some businesses present in the city's records do not appear in Yelp's data. In fact, it seems that no business of the "retail food" type appears in Yelp's data.

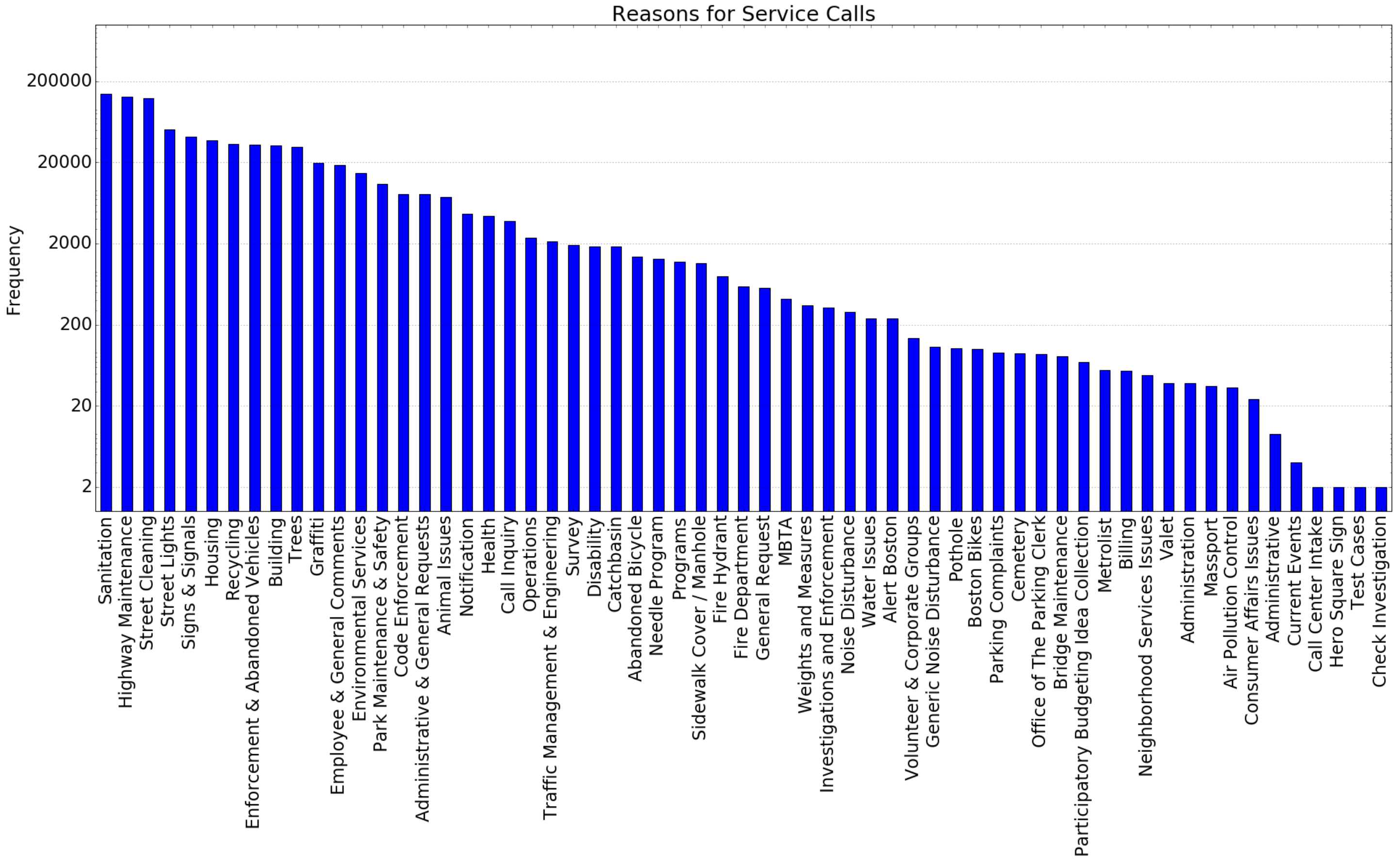
There are some important mistakes in Yelp's records. To see why, note for example that in Boston's records, the inspection of Samurai Kuang Eatery is marked as passing. For that inspection, four 1-star and one 3-star violations are noted, but this is only because that inspection was a follow-up to an inspection from one week before, which found those violations. Unfortunately, the Yelp dataset treats both inspections of the Samurai Kuang Eatery identically, marking four 1-star and one 3-star violations for each of the two inspections, which is quite misleading. This mistaken double-entry of health violations happens frequently throughout the Yelp dataset. We will see later that this issue, once we make an appropriate correction for it, will not impede the central purpose of this project.

In light of the comparison above, we can now consider the advantages of each dataset of inspection results. The City of Boston's dataset is kept continually up-to-date, with new results being entered as they occur, while Yelp's data ends in mid 2015. Boston's dataset is also more complete in the sense that each violation is categorized in much finer detail than a simple three-level severity rating. However, it should be noted that Yelp's dataset was developed with the express support of the City of Boston, working toward a purpose very similar to that of the present project. Most crucially, Yelp's dataset provides latitude and longitude coordinates for *every* inspection location listed. In this report, we use Boston's inspection data for some preliminary exploratory purposes, but all of our conclusive findings draw from Yelp's dataset instead.

Another quite useful dataset comes from the fact that Boston residents are able to dial "311" and report public property issues such as rodent sightings, unsanitary conditions, streetlight outages, and so forth, and records of these 311 service requests, from July 2011 to present, are available from data.cityofboston.gov. Each call record includes the following relevant information:

- The date the complaint was made
- Various descriptions of the nature of the complaint
- Latitude/longitude coordinates of the issue

Each complaint is described, often redundantly, by a "title," a "subject," a "reason," and a "type." In the dataset, there are 7837 different titles, 18 different subjects, 61 different reasons, and 215 different types. It is convenient to use "reasons" as a natural way to categorize complaints, since they strike a balance between being overly specific (as in the thousands of different "titles") and not being descriptive enough (like the handful of vague "subjects").



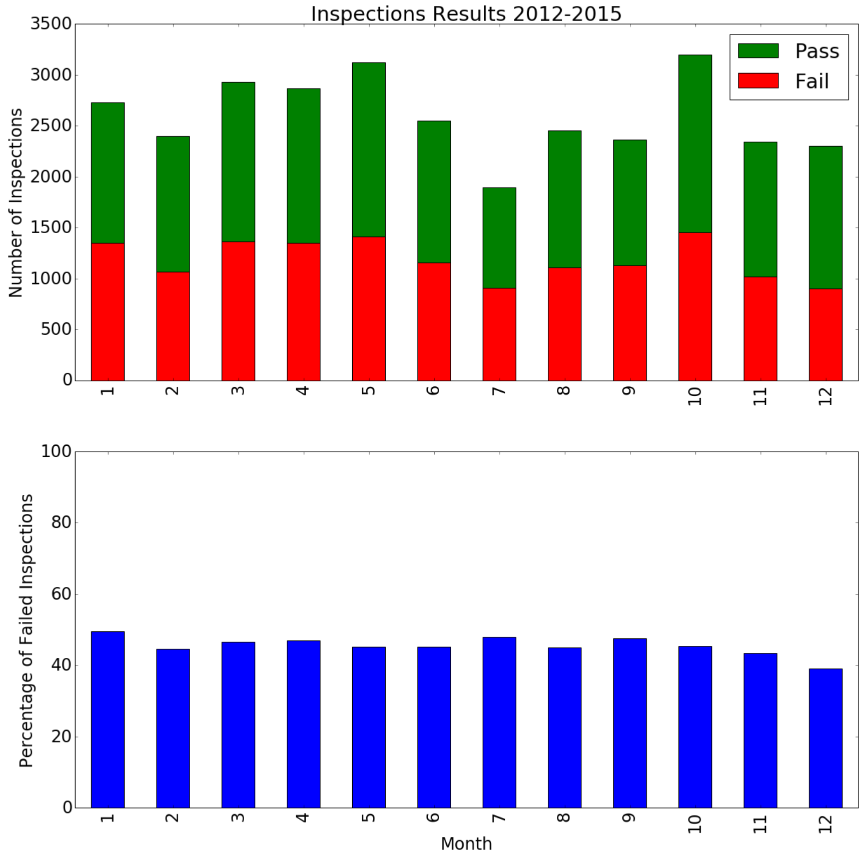
The preceding graph shows the prevalence of each of the various service call "reasons." Note that there is a swift decline in the frequencies of the least common reasons, with a handful of the very least common reasons only appearing in the dataset a few times. Because of this, in the work that follows we will disregard these extremely rare service reasons.

This 311 service calls dataset provides a wealth of details about reported environmental conditions near restaurant inspections. From these details we derive a model for predicting the outcomes of those inspections, later in this report.

2. Approach

Initial Data Exploration

Since service call data is only available from July 2011 onward, we will only consider inspection results from August 2011 onward. In Boston's inspection dataset, a total of 5604 businesses were inspected, with 3876 of them experiencing at least one failed inspection. Looking at the years 2012 through 2015, the number of inspections performed varies a great deal from month to month, but the *percentage of inspections that fail* is consistently between about 40 and 50 percent[*].

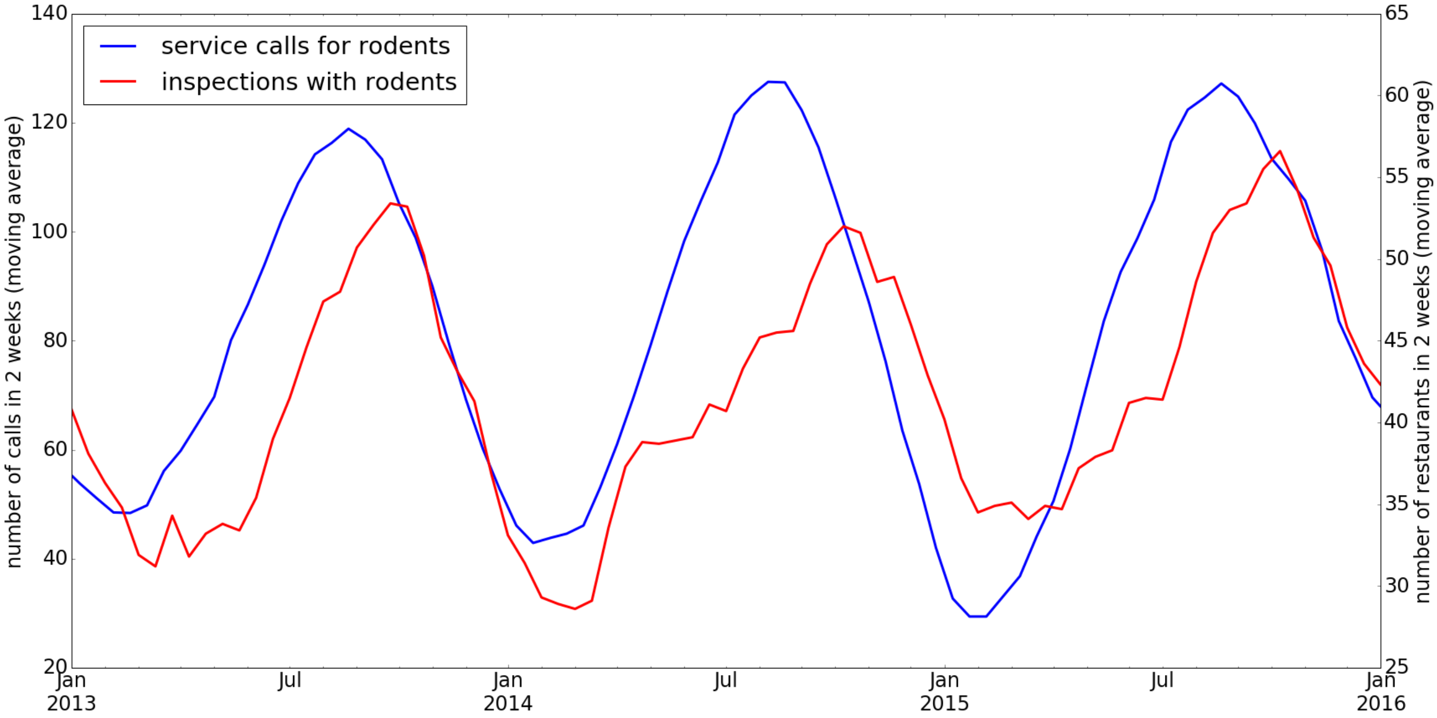


Our overall goal is to be able to predict the outcome (pass or fail) of a given health inspection, regardless of the specific underlying causes for a failure. However, our predictions are built upon environmental conditions near each inspected business, so it is worthwhile to consider some possible relationships between specific types of environmental issues and the reported causes of inspection failures. For instance,

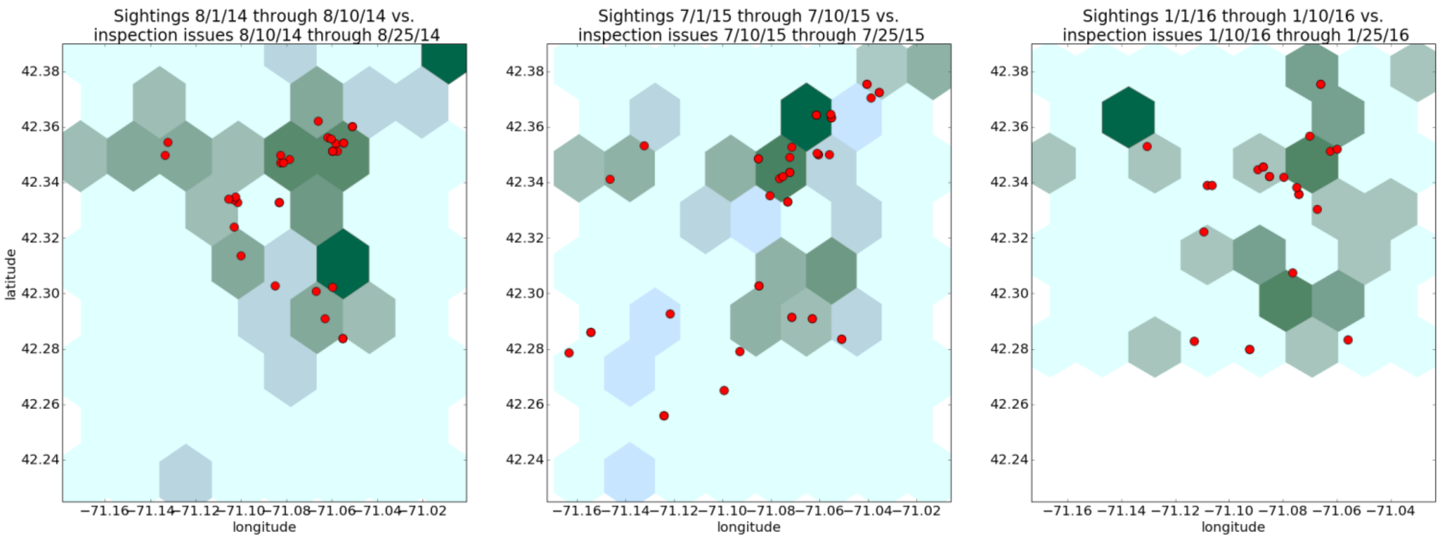
[*] It should be noted that in the more focused dataset provided by Yelp, the failure rate is about 64%.

one aspect of food safety that often inspires public interest is the presence of rodents in food service establishments.

The dataset reveals that over 20% of inspected businesses have experienced inspection failures related to rodents. Inspections with rodent-related issues are an interesting subset of all inspections because we can investigate, relatively easily, whether citizens' reports of rodent sightings (through 311 service calls) are good predictors of subsequent rodent-related issues during health inspections of nearby food service establishments. Intriguingly, based on service call data, rodents are reported via 311 most commonly around June to August, and are detected in health inspections most commonly shortly thereafter, in July to September:



The pattern above shows an unmistakable correlation between the *timing* of rodent sightings and rodent-related inspection issues. With more difficulty, we can use visuals to explore whether there might be a correlation between the *locations* of sightings and inspection issues. For the graphs below, three different 10-day periods are selected (more or less at random) and service calls involving rodent activity during those periods are mapped. This reported rodent activity is indicated in green, with darker green corresponding to more activity. Then, rodent-related health inspections during an immediately subsequent 15-day period are shown in red. In each case, the distributions of sightings and rodent-related inspection results do appear to be roughly similar, although this conclusion is admittedly subjective.

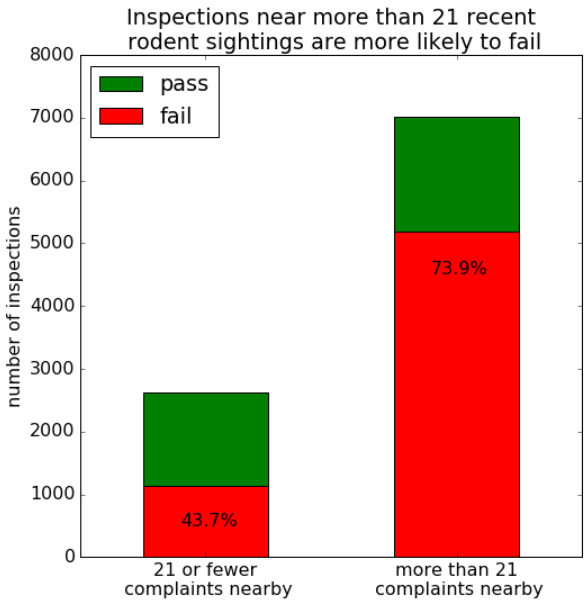

Sightings 8/1/14 through 8/10/14 vs.
inspection issues 8/10/14 through 8/25/14
Sightings 7/1/15 through 7/10/15 vs.
inspection issues 7/10/15 through 7/25/15
Sightings 1/1/16 through 1/10/16 vs.
inspection issues 1/10/16 through 1/25/16
latitude
42.38
42.36
42.34
42.32
42.30
42.28
42.26
42.24
−71.16
−71.14
−71.12
−71.10
−71.08
−71.06
−71.04
−71.02
longitude

Such patterns in both time and space lend some weak support to the idea that a restaurant that is near increased reported rodent activity might be at an increased risk of failing a health inspection. We will now see some firm statistical evidence that this is indeed the case.

Because we are interested in relationships between service call locations and restaurant inspection locations, for the remainder of this report, we will use only the inspection dataset provided by Yelp, which includes complete latitude and longitude coordinates for every inspection. Another somewhat less important reason for using this dataset will also be seen later. For each of the inspections recorded in the Yelp dataset, we count the number of times rodent-related issues (which are categorized under "Environmental Services") have been reported near the inspected business, between 5 and 15 days before the inspection. Nearness is defined by the distance in latitude/longitude coordinates; more specifically, we count any reports that have occurred within about 3 miles of the inspected business.

Since the question we wish to answer – will a given establishment fail an unannounced health inspection or not? – is binary in nature, we will test the relationship between the number of service calls and the proportion of failed inspections. In particular, do businesses that are near many "environmental services" issues tend to fail health inspections more frequently than other businesses? The first quartile for the number of environmental services complaints is 21, so let us perform a permutation test of the following hypotheses:

> **Null hypothesis.** Establishments where environmental services issues have been reported more than 21 times nearby, 5 to 15 days before an inspection, fail the health inspection at the *same* rate as establishments where those issues have been reported no more than 21 times.
>
> **Alternative hypothesis.** Establishments where environmental services issues have been reported more than 21 times nearby, 5 to 15 days before an inspection, fail the health inspection at a *different* rate than establishments where those issues have been reported no more than 21 times.



The result of this hypothesis test is that there is a statistically significant difference in the inspection failure rates of businesses that are near more than 21 recent environmental services complaints, compared to other businesses (*p*-value less than 0.001). In the Yelp dataset, these establishments fail inspections at a rate about 30% higher than others, as illustrated to the left.

Again, rodent-related issues are just one particularly interesting subset of public health challenges, serving here as a microcosm of the possible predictive relationship between service call data and inspection outcomes. For any other category of service call incidents, we may take the same approach as described above to test whether there is a statistically significant difference in inspection failure rates between restaurants near few recent complaints, and those near many recent complaints. The table below gives the results of those hypothesis tests for each of the service call categories.

To illustrate the meaning of this table, here is another example: Among all restaurant inspections, when we count the recent nearby service calls from the “catchbasin” category, the first quartile (Q_1) is 2 calls. The difference in inspection failure rates between businesses near more than 2 “catchbasin” calls and others is not statistically significant (*p*-value 0.270). This information is marked in **blue** in the table below.

category	Q_1	**difference in fail rates**	***p*-value**
Sanitation	244	0.283919	0.000
Highway Maintenance	225	0.398662	0.000
Street Cleaning	105	0.120231	0.000
Street Lights	127	0.064507	0.000
Signs & Signals	77	0.121797	0.000
Housing	74	0.146075	0.000
Recycling	36	0.403072	0.000
Enforcement	29	0.200246	0.000
Building	43	0.404795	0.000
Trees	22	0.535288	0.000
Graffiti	46	0.207686	0.000
Employee Comments	27	0.038716	0.001
Environmental Services	21	0.302576	0.000
Park Maintenance	7	0.252541	0.000
Code Enforcement	0	0.012661	0.236

category	Q_1	**difference in fail rates**	***p*-value**
Administrative Requests	14	0.005763	0.604
Animal Issues	0	0.067438	0.000
Notification	10	0.204228	0.000
Health	9	0.300819	0.000
Call Inquiry	0	0.242774	0.000
Operations	2	0.192627	0.000
Traffic Management	2	0.243865	0.000
Survey	0	0.085734	0.000
Disability	0	0.253455	0.000
Catchbasin	**2**	**0.012495**	**0.270**
Abandoned Bicycle	1	0.125026	0.000
Needle Program	0	0.214738	0.000
Programs	0	0.108116	0.000
Sidewalk Cover	1	0.132422	0.000
Fire Hydrant	0	0.076572	0.000

category	Q_1	**difference in fail rates**	***p*-value**
Fire Department	0	0.022308	0.216
General Request	0	0.160809	0.000
MBTA	0	0.268640	0.000
Weights and Measures	0	0.331160	0.000
Investigations	0	0.226512	0.000
Noise Disturbance	0	0.265195	0.000
Water Issues	0	0.273201	0.000
Alert Boston	0	0.132167	0.000
Volunteer Groups	0	0.232479	0.000
Noise Disturbance	0	0.125060	0.000
Pothole	0	0.241674	0.000
Boston Bikes	0	0.234363	0.000

category	Q_1	**difference in fail rates**	***p*-value**
Parking Complaints	0	0.177144	0.000
Cemetery	0	0.228341	0.000
Parking Clerk	0	0.207331	0.000
Bridge Maintenance	0	0.097254	0.000
PBIC	0	0.244861	0.000
Metrolist	0	0.227395	0.000
Billing	0	0.188847	0.000
Neighborhood Services	0	0.189732	0.000
Valet	0	0.187895	0.000
Administration	0	0.204654	0.000
Massport	0	0.223607	0.000
Air Pollution Control	0	0.186724	0.000

We can see in the table that for the vast majority of service call categories, there is a statistically significant difference in failure rates between establishments that are near many recent complaints and establishments that are not. This demonstrates that having knowledge about recent service calls that have occurred near a given inspection can provide a basis for predicting the outcome of the inspection.

Features and Predictive Modeling

In attempting to predict the outcome of a given restaurant inspection, we primarily look to conditions in the city near that restaurant in the recent past, as reflected in service calls. For each of the service call categories discussed before, we count the number of times that particular issue has been reported near the inspected business, between 5 and 15 days before the inspection. Additionally, for each business inspected we measure the number of days that have passed since its previous inspection, along with the results of the previous inspection, in the form of 1-star, 2-star, and 3-star violation counts.

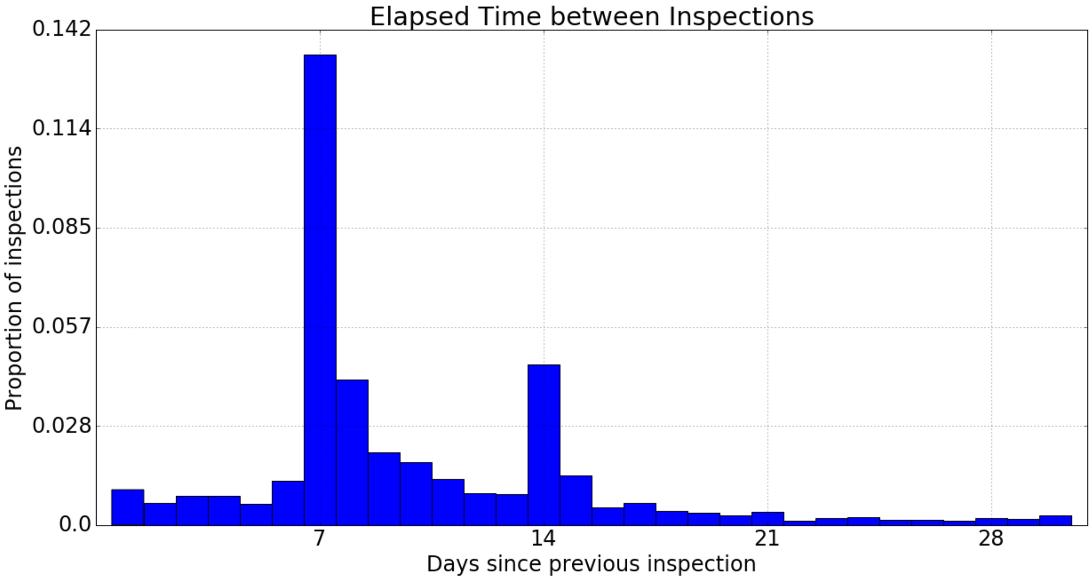
Below, for illustrative purposes, we display the extracted features and targets for some of the inspections that occurred on July 23, 2014. We see that four inspections occurred that day; one of these inspections passed (as indicated by a zero in the targets vector), while the others failed. Between 5 and 15 days prior to the inspection that passed, there were 759 complaints related to "sanitation" in the area around this business, 490 complaints related to “highway maintenance,” 300 related to “street cleaning,” and so on. It had been 182 days since the most recent inspection, during which zero violations were found at this business.

9641 data points

latitude	longitude	Sanitation	Highway Maintenance	Street Cleaning	Street Lights	...	Air Pollution Control	days since last inspection	violations last time			targets
									*	**	***	
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮	⋮	⋮	⋮
42.312	-71.114	769	500	297	118	...	0	196	0	0	0	1
42.361	-71.052	454	318	108	159	...	0	30	4	0	0	1
42.310	-71.115	759	490	300	117	...	0	182	0	0	0	0
42.361	-71.067	496	330	114	165	...	0	91	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮	⋮	⋮	⋮

60 features

As mentioned before, there are systematic mistakes in Yelp's dataset, with a kind of double-entry of health violations happening frequently throughout the dataset. While most inspections repeat after a period of about one year, we should be suspicious of any inspection results that supposedly took place within a few weeks of a previous inspection at the same business. With the "days since last inspection" feature of the data calculated (as seen above), we can take a closer look at such inspections. The graph below shows that quickly repeated inspections are overwhelmingly repeated after either exactly one week or exactly two weeks. Because of this, we make one final correction to the data by dropping any inspection with a calculated delay of 14 or fewer days.



We aim to devise a means for predicting the outcomes of future health inspections – will a given establishment fail an unannounced health inspection, or not? To this end, we have a matrix of 9641 data points and 60 features. Recall that of these features, 54 measure the recent and nearby occurrences of various kinds of city service complaints, while 4 of the features relate to the outcome of the restaurant's previous inspection, and 2 of the features are simply the latitude and longitude of the restaurant. We also have a vector of 9641 inspection outcomes, classifying each inspection as failed or passed.

To develop a model for classifying inspections as failing or passing, we will take the following general steps:

- Partition the features and targets described above into a training set and a test set. We randomly assign 80% of the data to the training set. The assignment is "stratified," meaning that the relative frequencies of failed and passed inspections are preserved in the training set.
- Use 3-fold cross-validation on the training set to search for hyperparameter values to maximize accuracy. More specifically, the training set is split into three parts. For each set of candidate hyperparameter values, each part of the training set in turn is held back while a model is created from the other two parts, then checked on the third. After all hyperparameter values are tested this way, the best-performing set of values are finally used on the full training data to create the final model.
- Use the previously untouched test set to check the model's accuracy (i.e. the proportion of inspection results that it classifies correctly), along with other more graphical measures of performance:
    - The ROC ("receiver operating characteristic") curve represents the set of all possible pairs of values for the model's true positive and false positive rates. In the context of our problem, the true positive rate is the proportion of failed inspections that are correctly identified by the model, while the false positive rate is the proportion of passed inspections that are incorrectly predicted to be failed by the model. A perfect model would have a true positive rate of 1.0 (every inspection that fails in reality can be predicted by the model) and a false positive rate of 0.0 (the model never predicts a failure for an inspection that passes in reality). The area under the ROC curve, or the AUC, would be 1.0 for a perfect model.
    - The precision-recall curve represents the set of all possible pairs of values for the model's precision and recall. In the context of our problem, the precision is the proportion of predicted inspection failures that are failed in reality, while the recall is simply a synonym for the true positive rate, explained above. A perfect model would have a precision of 1.0 (every inspection that is predicted to fail does fail in reality), and a recall of 1.0 as noted before. The area under the precision-recall curve for a perfect model would be 1.0.

These steps have been applied to create a variety of predictive models for restaurant inspections, using the concepts of logistic regression, individual decision trees, support vector machines, and tree ensemble techniques such as random forests and extremely randomized trees (also known as extra-trees classifiers). The table to the right gives an overview of the five main classifier types that have been applied to our problem, along with the best accuracy that each type of model typically achieves on the test data. As seen in the table, the performance of each type of model has been fairly comparable, but we will now detail the two most successful.

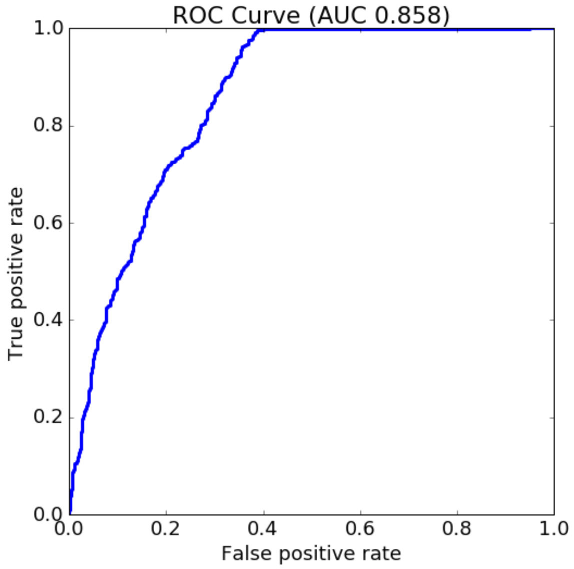
Classifier type:	**Typical accuracy:**
Logistic regression	0.83
Single decision tree	0.84
Support vector machine (SVM)	0.87
Random forest	0.88
Extremely randomized trees	0.86

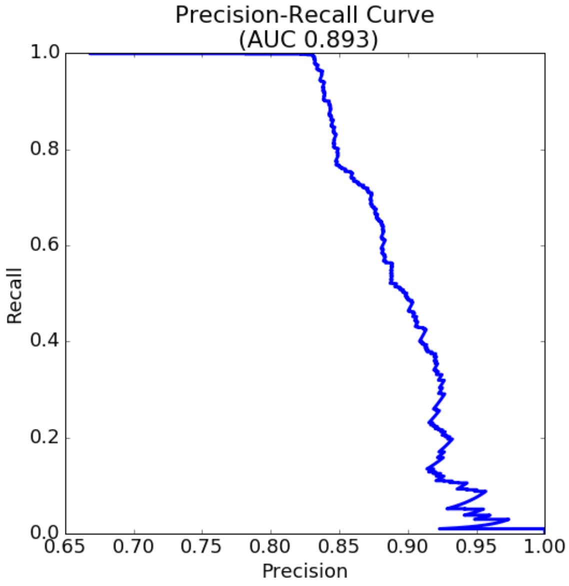
3. Findings

Support Vector Machines

Using the features and general steps described above, a support vector machine classifier can be constructed. We first standardize the feature values, then search for optimal values of the regularization hyperparameter, kernel type, and kernel coefficient. Without exception, a radial basis function kernel is found to be optimal for this problem. Once the model is created, its accuracy on our test set for this problem is typically around 0.87.

Here we have the ROC curve and precision-recall curve for a typical SVM classifier, along with the area under each curve (AUC):



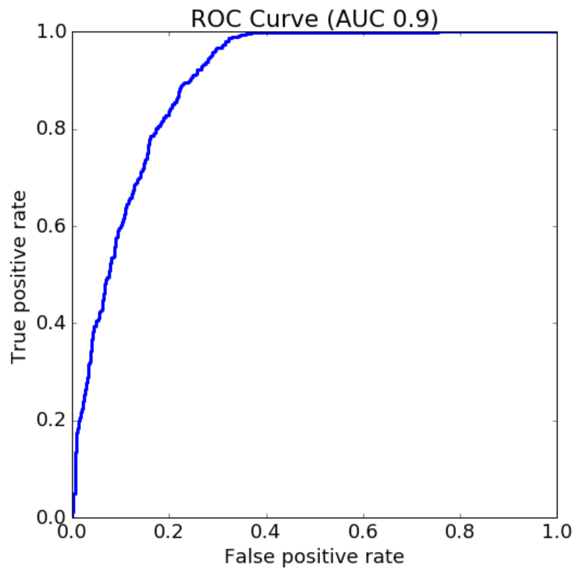


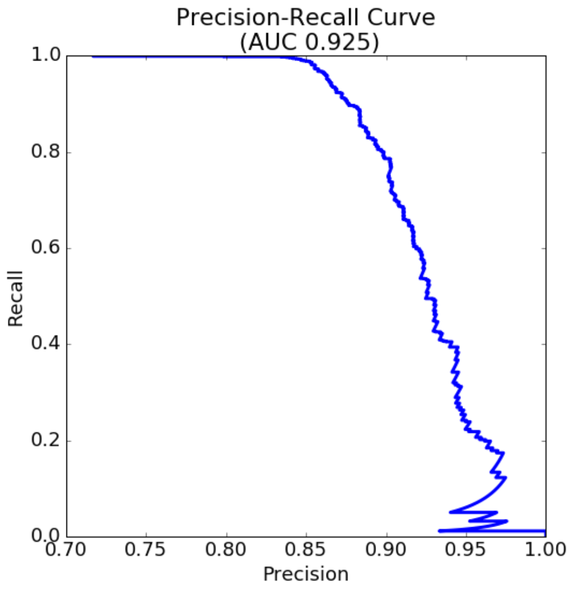
As noted before, a randomly-selected restaurant has a 64% chance of failing an unexpected inspection. However, the precision-recall curve shown here indicates that a restaurant identified by this SVM model as a predicted inspection failure will have an 83% or higher chance of failure. If desired, this precision can be increased further, at the cost of decreasing recall. For example, if the City is willing to accept only 50% recall (i.e. only 50% of failed inspections are foreseen), then based on the precision-recall curve seen above, the precision of predictions could be increased to about 90%.

Random Forests

Again using the features and general steps described before, we construct a random forest classifier. We search for optimal values of the following hyperparameters: the maximum depth of trees in the forest, the number of features to consider when splitting, the minimum number of samples allowed in a new leaf, and the minimum number of samples required when splitting a node. Once created, a random forest model typically achieves an accuracy of around 0.88 on our test set for this problem.

Below we have the ROC curve and precision-recall curve for a typical random forest classifier.

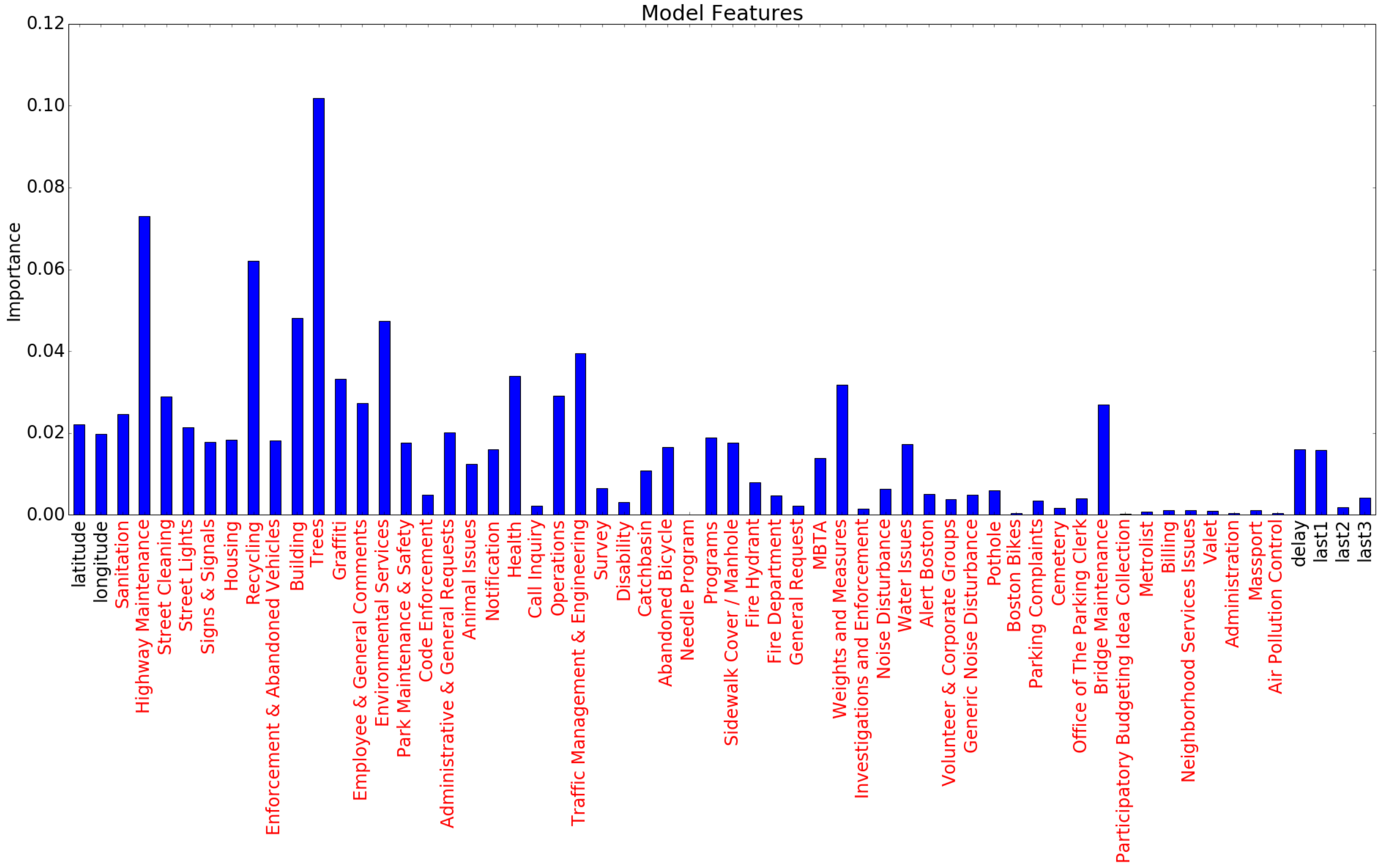




The precision-recall curve for this classifier indicates that a restaurant identified as a predicted inspection failure will have an 84% or higher chance of failure. Here too, if desired, this precision can be increased, with decreased recall. Based on the precision-recall curve, at 50% recall, the precision of predictions could be increased to about 93%. By every measure, this random forest classifier slightly outperforms the SVM classifier.

The use of a tree-based classifier provides a straightforward way to measure the importance of the model's features. For the random forest with the results shown above, the most important features were the service call reasons of "Trees," "Highway Maintenance," "Recycling," "Building," and "Environmental Services." The features that are ranked as the most important in one random forest model may vary somewhat when a new forest is constructed – even if the same data and hyperparameters are used – but in the present situation, the set of most important features is fairly consistent from one forest to another.

In the graph below, we visualize the importance of all 60 features used, as revealed via our random forest model. Note that other than the first 2 features on the left, and the last 4 features on the right, the rest of the features (labeled in red) are related to service requests. Note also that these features are listed in order of decreasing frequency (just as they were listed in a previous graph). Unsurprisingly, we can see a slight tendency for the more prevalent service call reasons to have higher importance than the less prevalent ones.



Comparison to Previous Contest Results

As mentioned before, the Yelp dataset that has been used here, primarily as a target for classifiers, was released as part of the "Keeping it Fresh: Predict Restaurant Inspections" contest at DrivenData.org in 2015. The goal of that contest was to be able to predict the number of one-star, two-star, and three-star violations found at each health inspection during a six-week period after the closing of contest submissions. So, in contrast to our present goal of simple classification, the DrivenData contest involved a problem of regression. Contestants' predictions were judged using the following weighted root mean square log error formula, with lower error scores corresponding to better performance:

$$\text{WRMSLE} \;=\; \sqrt{\frac{1}{N}\sum_{i\,=\,1}^{N}[\log(y_i \cdot W + 1) - \log(\hat{y}_i \cdot W + 1)]^2}$$

In the notation used for this scoring function, there are N inspections, y_i represents the results (a vector of three integers) of the i^{th} inspection, W is the weighting vector $(1, 2, 5)$[†], and $\hat{y}_i$ represents the predicted results of the i^{th} inspection.

Contestants extracted features mainly from the text of customer reviews and other information provided through Yelp, in addition to details from past inspection outcomes. In the present project, we have taken a very different approach, using features related to reported environmental conditions near upcoming

[†] In plainer terms, contestants' predictions for the number of one-star, two-star, and three-star violations at each inspection were weighted so that the two-star prediction was worth twice as much as the one-star prediction, while the three-star prediction was worth five times as much as the one-star prediction.

restaurant inspections. It would be interesting to compare the predictive ability of our features with those used in the contest. To make a somewhat fair comparison, using the same set of 60 features upon which we built our previous classification models, we take the following steps:

- Set aside the last 6 weeks of available data as a test set, with the rest of the data as a training set.
- Use cross-validation on the training set to search for hyperparameter values to minimize the WRMSLE score, and use those best values to create a regression model.
- Use the previously untouched test set to calculate the regression model's WRMSLE score.

A random forest regression model created in this way typically achieves a WRMSLE score of around 0.85 on the test data – a level of error quite comparable to the leading results of the DrivenData competition, where the winning contestant's error score was 0.8901.

4. Possible Directions for Future Study

The environmental features we have extracted provide enough information for a successful classifier, but there is likely room for improvement. We have used the same time window and nearness threshold in the calculation of each of those features. Perhaps some variation in these choices from feature to feature would lead to a better-performing model, since some environmental factors might have influence at different distances, or on different timescales, than others.

If the City of Boston's more detailed inspection records are used in place of Yelp's dataset, we may incorporate new features drawn from the specifics of past inspections, such as the exact reasons for past failures. After all, it seems plausible that some health issues are more prone to repeat offense than others.

Lastly, given the success of the DrivenData contestants in using customer reviews to predict inspection outcomes, our model might benefit from an incorporation of features derived from textual data provided by Yelp and Google restaurant reviews. We have made some preliminary attempts in this direction, by using both the fairly unsophisticated "bag of words" and "term frequency-inverse document frequency" (TF-IDF) approaches on the reviews text, followed by unsupervised dimensionality reduction, to produce a more extensive set of model features. As of now, these attempts have not yielded a noticeable improvement in predictor performance, and so the fine details of that preliminary work are not included here.

5. Recommendations

Given our successful development of an accurate predictor for health inspection failures, the City of Boston is advised to make use of such a predictor in planning upcoming inspections, in order to more efficiently utilize inspectors' time.

As noted before, the predictor's precision (the fraction of predicted inspection failures that turn out to be failures in reality) can be increased well above 90%, at the cost of decreasing its recall (the fraction of failures that are foreseen by the predictor). This trade-off should be calibrated in consultation with the Inspectional Services Department of Boston.

Lastly, the models developed in this project have been based on a limited record of past health inspection results, for the principal reason that the City of Boston's live records lack latitude and longitude information for many inspections. As such, our models can be seen as a proof of concept; a fully operational model for ongoing use would need to incorporate the City's live records. Therefore the City is advised to require all health inspections to include location data. Alternatively, missing location data for past and future inspections may be generated rather inexpensively through the use of the Google Maps Geocoding API. The code included with this report already gives some indication as to how this would be done.