DS-GA 1003 Spring 2017
Final Project Report

Predicting New York City Restaurant Food Safety Violations

Jonathan Toy (jt2276)
Julie Helmers (jch609)
Seda Bilaloglu (sb3923)

Project Information

This project was done as the final project for the DS-GA 1003 Machine Learning and Computational Statistics course in the Center for Data Science at NYU. We were advised by Dr. Bonnie Ray. Our code and data files, along with instructions on how to replicate our results, can be found in our GitHub repository at https://github.com/jchelmers/urban-data-project.

Introduction

There are approximately 24,000 food establishments in New York City, each of which must be inspected by the Department of Health and Mental Hygiene (DOHMH) at least once a year [11]. The department's food safety inspectors therefore face an average of 96 restaurants per business day. They also face the decision of which restaurants to visit on any given day: Each establishment is on a re-inspection cycle ranging from 3-5 months to 11-13 months depending on its most recent inspection score [11]. Ideally, each day the DOHMH inspectors should visit the restaurants that are within the re-inspection window and have the highest risk of committing dangerous food safety violations. With this goal in mind, we mined data from DOHMH, 311 Services, Department of Consumer Affairs, New York State Liquor Authority, Weather Underground, OpenStreetMap, and Google Places and classified New York City restaurants according to their probability of exhibiting two or more critical food safety violations on a particular date. We drew inspiration from the work of the City of Chicago and the startup Open Data Nation, both of which have experienced success in helping other cities tackle similar machine learning problems to more efficiently dispatch food safety inspectors [9, 10]. After feature engineering, parameter tuning, and examining ROC curves, our winning model was a gradient boosted classifier with the deviance loss function, 0.4 bag fraction, regression trees with unrestricted feature selection, and 120 boosting steps. With this model and a classification threshold of 31.5%, we were able to correctly classify 75% of the restaurants with 2+ critical violations on their most recent inspection, while only misidentifying 44% of the restaurants that had zero or one critical violations.

Datasets and Preprocessing

Primary Dataset

Our primary dataset was the New York Department of Health and Mental Hygiene (DOHMH)'s New York City Restaurant Inspection Results [3]. The dataset is automatically updated daily. As of when we downloaded it on March 22, 2017, it had 424,888 rows and 18 columns. Each row corresponds to a single violation (or lack thereof) for a New York City food establishment. The violations are classified as either "critical" or "non-critical" according to the DOHMH's criteria. Broadly speaking, non-critical violations are either administrative in nature or indicate that the proper steps to prevent a potential problem have not been taken, such as inadequate rodent-proofing, while critical violations indicate evidence of a current problem, such as evidence of rodents. Because our aim is to optimize the DOHMH inspectors' priorities to best protect the safety of the public, we are most interested in critical violations.

The columns of the original dataset are described in detail in the table in the Appendix, which is taken from the DOHMH's overview of the data. The dataset includes some missing values, in particular missing scores, grades, grade dates, and violation types and descriptions. Some of these are not truly missing but are not applicable for a given row, for example if a grade was not assigned at the time or no violations were found. The dataset also includes some rows with an inspection "date" of January 1, 1900, which correspond to restaurants that have not been inspected yet.

We preprocessed the restaurant inspection results by removing any food violations that we deemed irrelevant to food safety, namely the non-critical violations related to calorie counts, display of no smoking policy information, and trans fat regulations. We also removed any violations that were found on "administrative" inspections and "non-operational" inspections that were conducted while a restaurant was not in business. Then we used the CAMIS id, a unique identifier for each restaurant, to group all the violations belonging to a single establishment. We excluded any restaurants that had zero or one inspections, giving us 22,330 remaining restaurants.

For our target variable, we calculated the number of critical violations on a restaurant's most recent inspection. For our predictor variables, we retained each restaurant's cuisine type, zipcode, borough, and address (later removed but used for further feature generation, below). In addition, we computed the following features from each restaurant's violation history:

- Number of previous inspections
- Time in days since last inspection
- Time in days since first inspection
- Total number of critical violations on previous inspections
- Total number of non-critical violations on previous inspections
- Average number of critical violations on previous inspections
- Average number of non-critical violations on previous inspections
- Number of critical violations on second-most recent inspection
- Percentage of previous inspections with at least two critical violations

Importantly, we separated out any data corresponding to a restaurant's most recent inspection when extracting these features (other than time since last inspection). Because the number of critical violations on a restaurant's most recent inspection was our target variable, we needed to avoid information leakage by making sure that the results of that inspection were not taken into account in the features described above.

Secondary Datasets

To improve our predictions of critical food safety violations, we integrated the other datasets listed below to mine additional predictor variables. We chose these datasets based on features used by the City of Chicago and Open Data Nation [9, 10] and our own intuitive hypotheses about factors that might either affect food safety conditions, such as the weather or the presence of rodents, or reflect food safety conditions, such as Google star ratings or reports of food poisoning.

311 Service Requests from 2010 to Present

We extracted several predictor variables from New York City Social Services' 311 Service Requests from 2010 to Present [1]. We filtered the 311 calls to only include calls with the complaint type of "Food Establishment," "Food Poisoning," "Rodent," "Dirty Conditions," "Missed Trash Collection," "Electricity," "Safety," "Police Matter" and "General." These types were chosen to explore our hypothesis that relevant 311 complaints about or near a restaurant may predict that restaurant's likelihood of having critical food safety violations. As of when we last accessed the 311 dataset on March 22, 2017 (it is also updated daily), when filtered for the last year, there were 31598 rodent, 33837 dirty conditions, 27509 electric, 9811 food establishment, 25772 general, 30306 missed collections, 55468 noise, 8868 police matter, 8000 safety, and 3067 food poisoning complaints . The dataset has 53 variables. We were only concerned with the X and Y state plane coordinates of the incident location, type of complaint, and the date that the service request was created.

The latitude and longitude of each restaurant's location were calculated from the address by using OpenStreetMap's search engine, Nominatim [8]. Nominatum was not able to convert approximately 10% percent of the addresses, so for those we used the Google Maps Geocoding API [6]. (We did not use the Google Maps API for the entire dataset because of its 2500 daily request limit.) Google Maps API couldn't convert the addresses of 27 restaurants, mostly due to inaccurate address entry, so we excluded those restaurants from our analysis. Then, for each restaurant, we calculated the number of these complaints of each type that were placed within 0.3 km (about 3 streets) within the last year and added those counts as features to our dataset.

Historical Weather Data

We integrated historical weather data at the zipcode-level from the Weather Underground Weather History and Data Archive [5]. We suspected that higher temperatures and higher humidities would lead to more food safety problems overall and might prove particularly challenging to restaurants with riskier food safety practices. For each of our restaurants, we used Python's BeautifulSoup package to scrape the temperature and humidity from the Weather Underground weather station in that zipcode on the date of the most recent inspection and the two days prior. If the weather data was not available, for example if there was not a weather station in that zipcode as of that date, we used the temperature and humidity from the Central Park weather station instead. We were missing temperature and/or humidity data for 1854 restaurants, or about 8% of our data. We then incorporated the three-day average temperature (in degrees Fahrenheit) and three-day average humidity (as a percentage) leading up to the most recent inspection date as features in our dataset.

Liquor and Sidewalk Cafe Licenses

We also included predictor variables that indicated the type, if any, of sidewalk cafe license or liquor license held by each restaurant. We added this information because we hypothesized that establishments with outdoor eating areas (sidewalk cafes) might be more susceptible to food safety problems, and that very crowded establishments like bars, which tend to have liquor licenses, might also struggle more with food safety. We retrieved the license information from the New York State Liquor Authority's Quarterly List of Active Licenses and the Department of Consumer Affairs (DCA)'s Sidewalk Cafe Licenses and Applications [4, 2]. Both datasets contained a much higher level of detail about the licenses than we needed; we simply matched restaurants with the relevant sidewalk and/or liquor licenses based on latitude and longitude, with some leeway for error to allow for incorrectly entered or imprecise location data. We then used the liquor license category and sidewalk license category – including "N/A" for no active liquor license or sidewalk license, respectively – as additional features.

Google Places Business Ratings

We obtained the average star rating (on a scale of 0 to 5) of each unique restaurant establishment through text search queries to the Google Places API based the restaurant's listed address. We theorized that restaurants with higher star ratings would be less likely to have many critical food violations on the next inspection. The Google Places API [7] was unable to retrieve business ratings for approximately 6% of the dataset, primarily in cases where those restaurants had never been rated by Google users. In these instances, ratings were imputed using the average business rating in the associated zipcode if possible, and using the average business rating in the dataset if the zipcode data was not available. Indicator variables for each type of missing ratings (missing at the restaurant level or missing at the zipcode level) were included as features along with the average star rating.

Final Preprocessing

After generating the features from the datasets above, we removed the address and latitude and longitude variables and normalized the remaining numerical features to fall in the interval [0,1]. We encoded the categorical features with k categories as k-1 binary variables; the kth level that was dropped was chosen arbitrarily. Finally, we randomly split our 22,330 restaurants into a held-out test set (20% of the data) and a training set (80% of the data), on which we later did cross-validation.

Performance Evaluation

As mentioned above, we withheld information related to the most recent food safety inspection for each restaurant and tried to predict the occurrence of two or more critical violations on that as our target variable. As there are merits to both viewing this as a classification problem (i.e. generating a list of restaurants that may be violating health codes) and as a ranking problem (i.e. identifying the worst offending restaurants), we considered both classification metrics (accuracy, recall, precision and F1-score) and ranking metrics (AUC score and ROC curves) to evaluate performance. Because the AUC and ROC ranking metrics are independent of the classification threshold – the threshold at which examples are classified as likely to have at least two critical violations – while the classification metrics vary wildly depending on the threshold, we ultimately prioritized the AUC score and a visual inspection of the ROC curves when choosing between our models.

Baseline Algorithm

Approximately 66% of our 22,330 restaurants had fewer than two critical violations on their most recent inspection, so we chose to predict the majority class, i.e. zero or one critical violations, as our baseline algorithm. This baseline results in an AUC score of exactly 0.5. (It's interesting to note that we chose two or more critical violations as our target variable because almost all of the restaurants had at least one critical violation on their most recent inspection, and we wanted to have more balanced classes in our training set.) We also considered a baseline model of a naïve Bayes model trained only using the features from the primary dataset.

Modeling and Performance

In order to improve our performance beyond the baseline, we focused primarily on engineering new features from secondary datasets, as outlined above, and on introducing and tuning new model types. We experimented with logistic regression (LR) with both L1 and L2 regularization, naïve Bayes (NB) with both fit and uniform priors, random forests (RF) with various numbers of trees, and gradient boosting classification (GBC) models. We used the scikit-learn implementations of each. As there was a sizeable gap in performance between the gradient boosting and random forest models and the remaining models, we chose to focus on tuning the performance of these two model types in particular.

Gradient Boosted Classification Tuning

For the gradient boosted classification model, the main hyper-parameters that we tuned were the loss function, the bag fraction used for computing the gradient step, the number of features considered at each node split of each regression tree, and the total number of boosting steps. By measuring changes in mean AUC score on validation data using cross-validation, we determined that the optimal gradient boosted classifier uses deviance as a loss function, uses 40% of the sample to compute the gradient, uses regression trees that have unrestricted feature selection, and has 120 boosting steps. This final gradient boosted classifier model yielded AUC scores of approximately 0.691 (+/- 0.012) on validation data. Figures 1-4 show more detail on our parameter tuning.

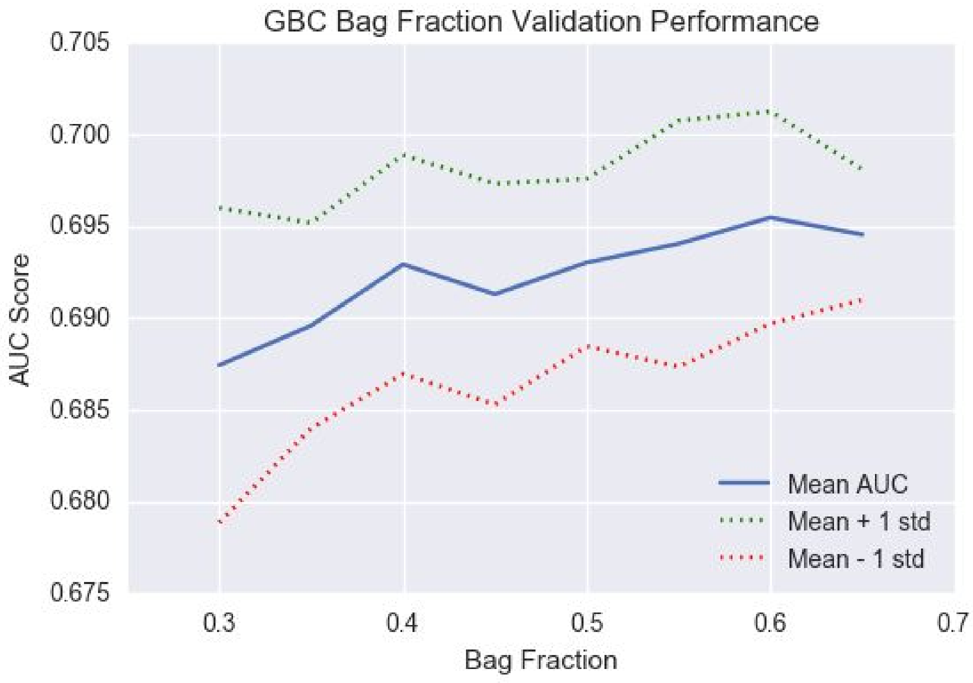

Fig. 1: As performance plateaus at a bag fraction of 0.6, we chose the lowest bag fraction with mean performance within one standard deviation of this, which was 0.4.

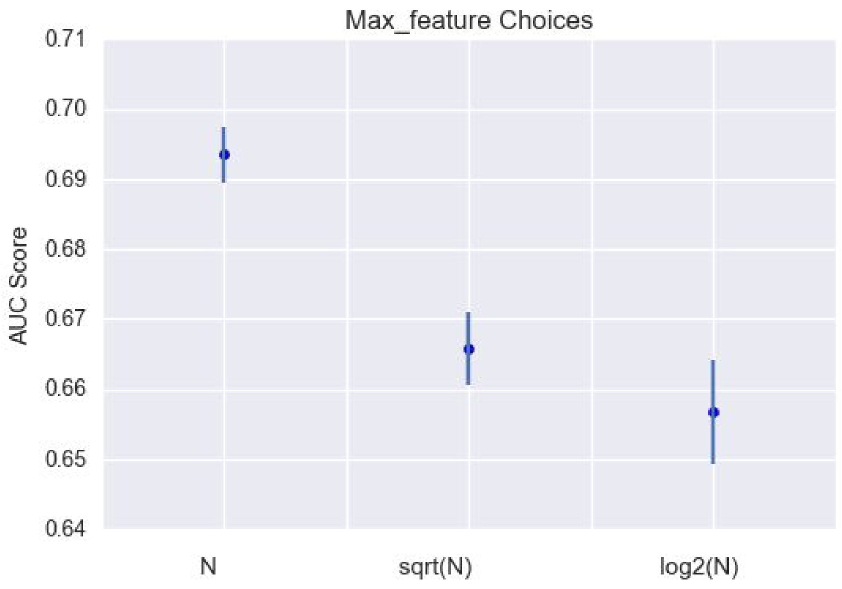

Fig. 2: The default option of including all features when making regression tree splits is by far the optimal choice based on the AUC.

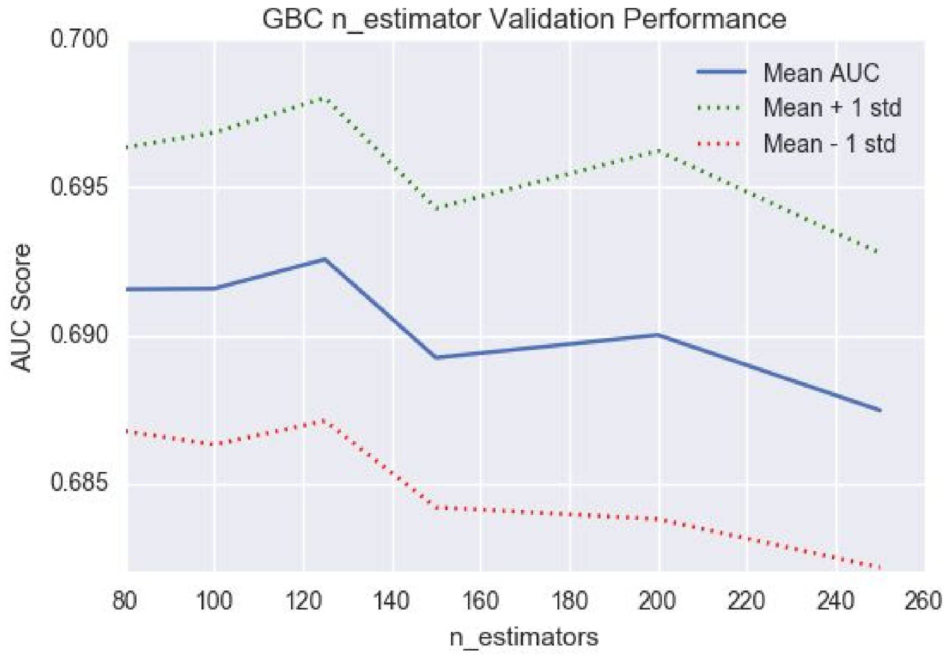

Fig. 3: As performance appears to increase up to 120 estimators and drop off afterwards, we chose this to be the optimal choice for the number of estimators.

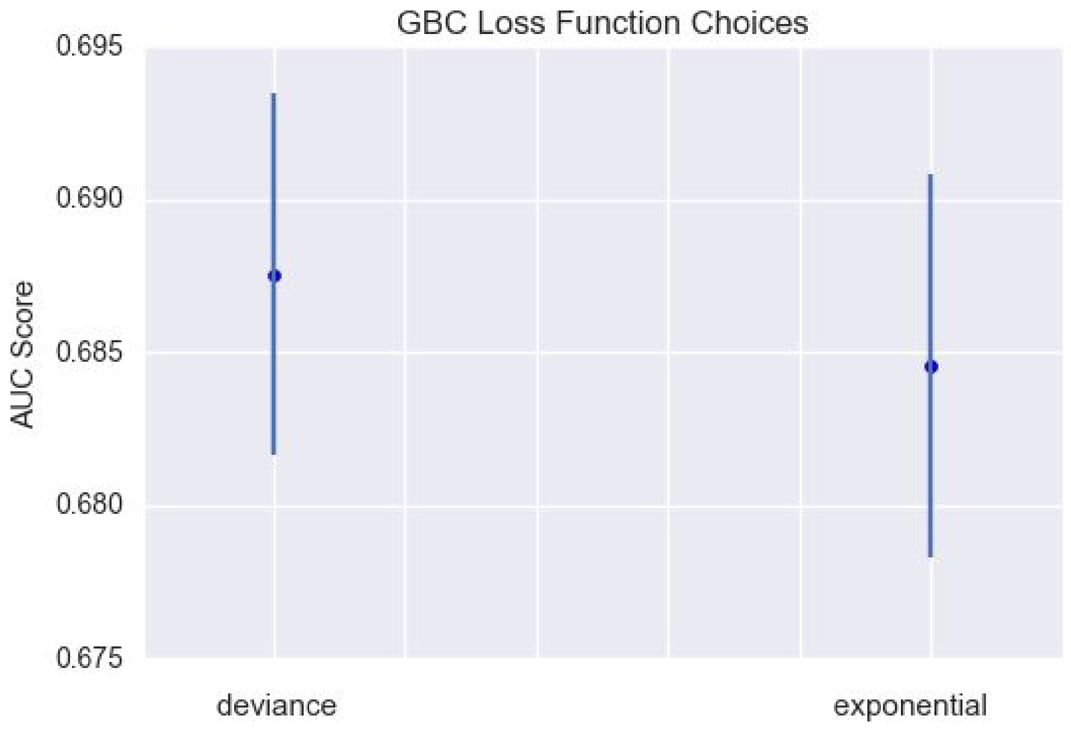

Fig. 4: Performance of the two loss functions are within one standard deviation of one another, so we chose the default option of deviance for the model.

Random Forest Classification Tuning

For the random forest model, the main hyper-parameters that we tuned were the number of features considered at each node split and the total number of trees in the forest. As the ensemble nature of the model helps prevent the forest from overfitting the data regardless of whether the individual trees overfit, the depth of the trees was not restricted. By measuring changes in mean AUC score on validation data, we determined that the optimal random forest would have unrestricted feature selection at nodes and 200 trees. This final random forest model yielded AUC scores of approximately 0.679 (+/- 0.005) on validation data. Figures 5-6 show more detail.

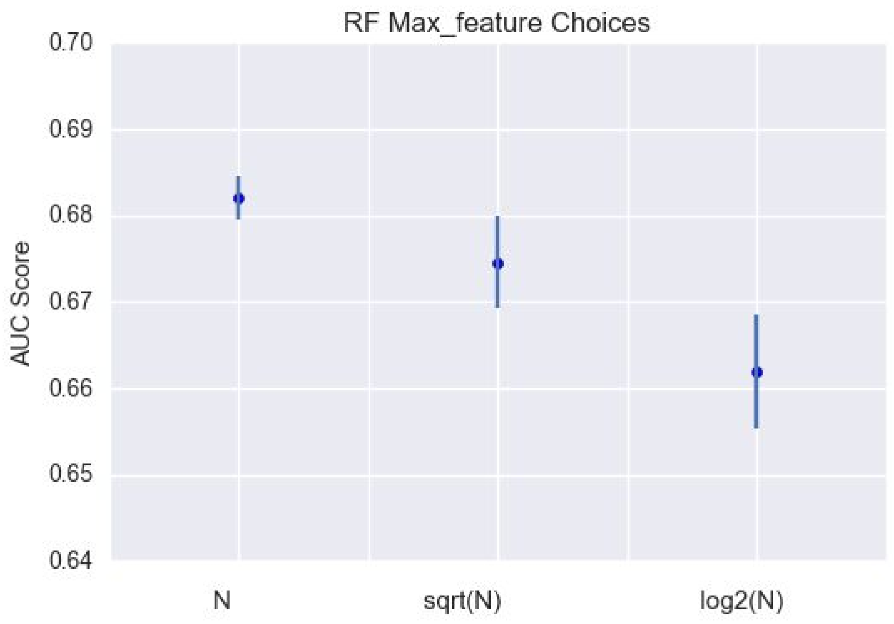

Fig. 5: The performance of the default option to include all features when considering node splits is just outside 1 standard deviation of the next best option, making it the optimal choice.

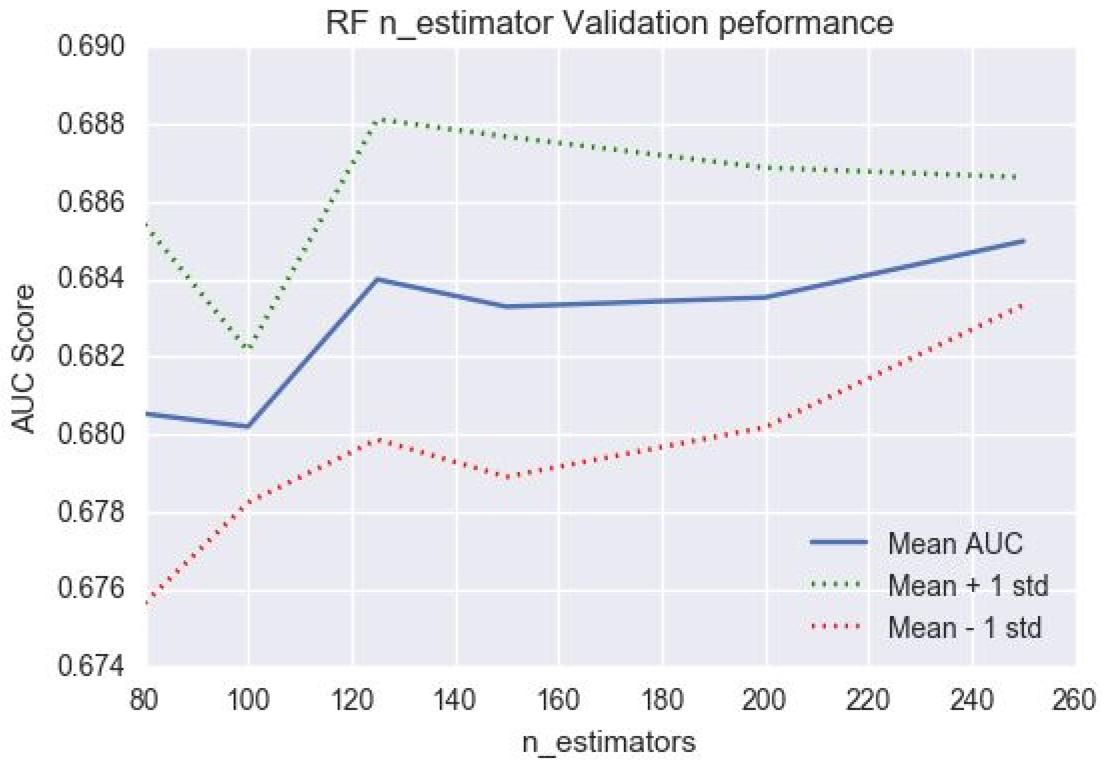

Fig. 6: Performance appears to increase with choice of n_estimators. A limit of 200 was chosen due to computational constraints.

Winning Model

As shown by the ROC curves in Figure 7, the tuned random forest (RF-200T) model was globally optimal to the best naïve Bayes (NB-UniPr) and best logistic regression (LR-L2) models when evaluated using cross-validation on our training set. However, the gradient boosted classifier (GBC) was optimal to the random forest model for all classification thresholds except those with false positive rates below 0.25. As we are more concerned with identifying restaurants in violation of important safety regulations (i.e. finding true positives) than with avoiding flagging restaurants that are not high-risk (i.e. avoiding false positives), we are more interested in other parts of the ROC curve. Thus, we prefer the gradient boosted classifier. The table below Figure 7 summarizes the AUC scores of our top models.

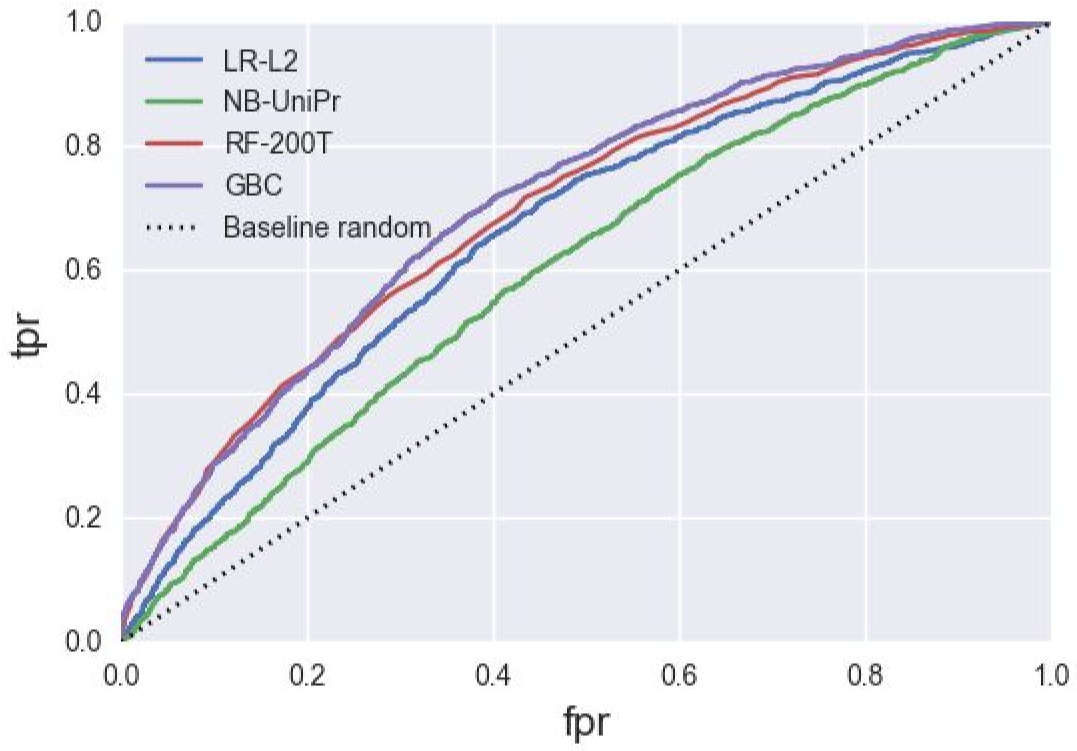

Fig. 7: The gradient boosted classifier (GBC) outperforms all other models in all areas of the ROC curve except the bottom left corner, i.e. where there is a very low false positive rate but, more relevantly to our goals, also a very low true positive rate.

	Mean AUC	**Std of AUC**
Baseline Majority	0.5	0
Baseline NB	0.567	0.016
Final NB	0.589	0.011
Final LR	0.646	0.015
Final RF	0.679	0.005
Final GBC	0.691	0.012

Feature Comparison

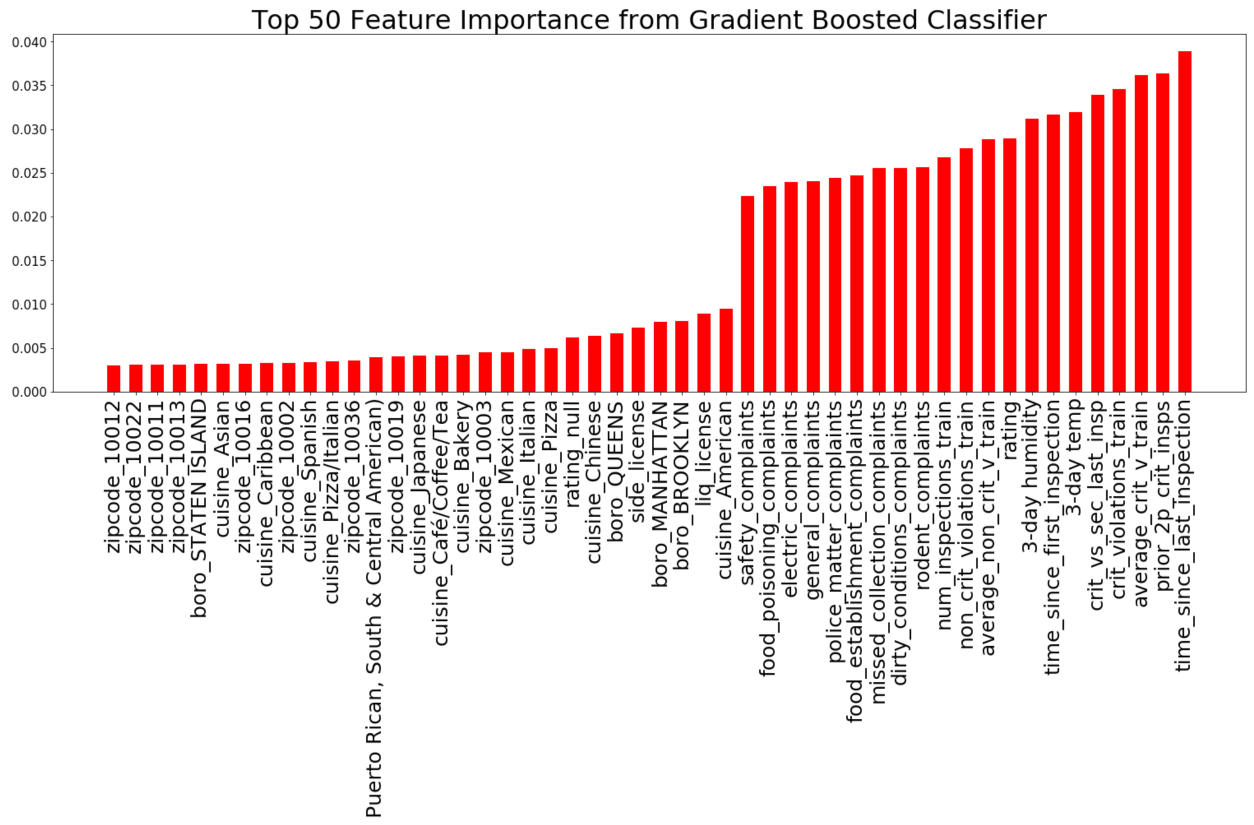

Fig. 8: These 50 features were most important in determining the predictions of our top model (the gradient boosting classifier).

To better understand our results, we plotted the importance of the features according to the winning gradient boosted classifier model. The top 50 features, seen in Fig. 8, reveal that the variables drawn from violation/inspection history, like number of previous inspections, time since last inspection, average number of critical violations, etc. contributed more to our predictions than other features. The predictive variables that we engineered from outside datasets, like three-day temperature/humidity and Google star rating, showed less importance than the primary variables. Notably, the 311 service request features appear among the top features but contributed less to the predictions, and are clustered together. The reason why the complaints features are very similar in importance might be that they are all associated with a certain location and time frame, and hence they might be highly correlated. The liquor and sidewalk cafe license features showed less importance, followed by the zipcodes, boroughs, and and cuisine type categories. Some cuisines, such as American and Chinese, and some zipcodes, such as 10003 (around East Village), contributed more to our predictions than other levels of those categories, due to their increased share of restaurants that two or more critical violations. Figure 9, for example, shows the breakdown of cuisine types according to how many restaurants had two or more critical violations.

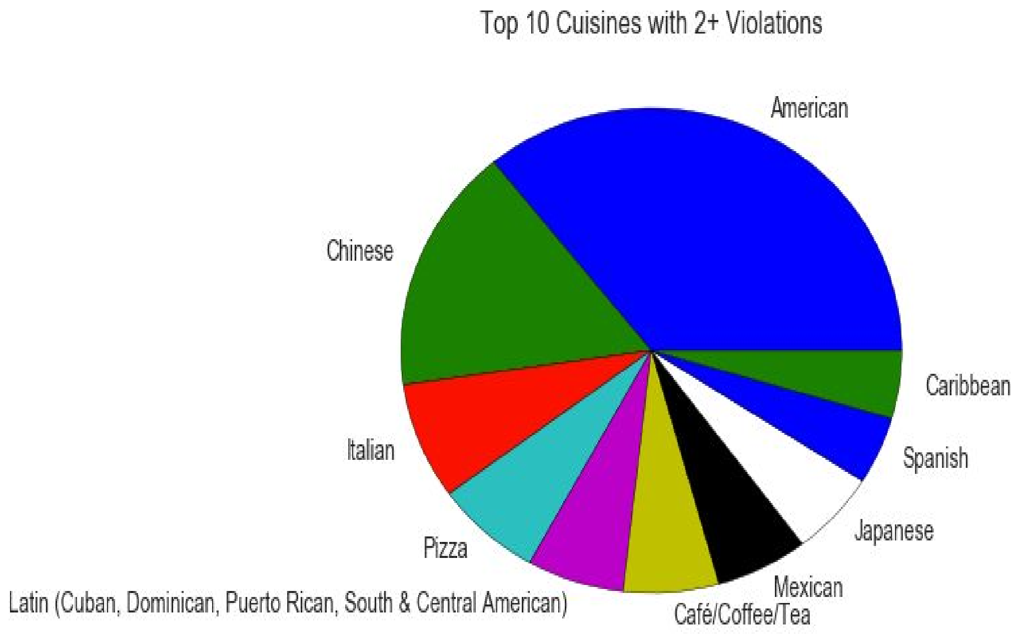

Fig. 9: These 10 cuisine types had the highest number of restaurants with at least two critical violations on their most recent inspection. For example, the American category had the most “offending” restaurants, followed by the Chinese category.

Challenges, Limitations, and Future Work

The main challenge we faced was that our engineered features were computationally costly to obtain, as they required crawling a large number of web pages, and some were less important than we expected (namely the 311 service requests). We also encountered several data quality issues in these additional datasets, including missing values for weather and Google ratings, which we had to impute, and inaccurate address information for some restaurants, which led to unsuccessful retrievals of latitude and longitude and ultimately exclusion of those records. In addition, there were multiple datasets that we had hoped to incorporate but were not able to. We had planned to include the restaurant size and the number of employees, as we hypothesized that they would improve the accuracy of our predictions – understaffed restaurants seemed likely to have more critical food safety violations. However, we couldn’t find any publicly available datasets with this information. Furthermore, we had wanted to include an indicator for the DOHMH inspector’s identity, as previous work in the city of Chicago has shown that this can be a strong predictor of the inspection outcomes [9]. Unlike Chicago, New York City does not publicly identify the inspector associated with each inspection, even by anonymous ID, and thus we couldn’t take that into consideration. Finally, we suspected that customer reviews, ratings, and restaurant price range scraped from TripAdvisor and Yelp could help predict food safety issues, but neither TripAdvisor nor Yelp publishes their New York data, and both companies restrict scraping.

For future extensions of this work, we feel that any of the above features, should the relevant data become available, could improve performance significantly. There are a number of other public datasets that we did not have time to investigate but that may help as well, including reports of nearby crimes, construction permits, and taxi trips to/from the restaurant.

Conclusions

Our goal is to identify restaurants that are likely to be violating health codes on a given date, so that DOHMH inspectors can prioritize visiting those restaurants. Thus, we want to maximize the true positive rate of our top model, and we should choose a classification threshold that allows for a high true positive rate while still maintaining a reasonable false positive rate: Some false positives are acceptable, because any restaurant that is within its re-inspection window (and therefore a candidate to be fed into the model) will need to be inspected at some point, but we are cognizant of the limited number of DOHMH inspectors working on any given day. In particular, we can see from the ROC curves that if we aim to achieve a true positive rate of at least 75%, we should set the classification threshold to be 31.5% rather than the standard 50%. If we do so, our winning gradient boosting classifier will only flag 44% of “clean” restaurants (false positives), and 55% of restaurants overall. Figure 10 shows the flagged restaurants as well as the false negatives on the map of New York City. If health inspectors were to utilize this model and prioritize visiting restaurants according to their predicted probability of receiving 2+ critical violations, they would be able to catch 75% of offending restaurants while still retaining 45% of their allotted time to catch the remaining 25% of high-risk restaurants that the model failed to identify.

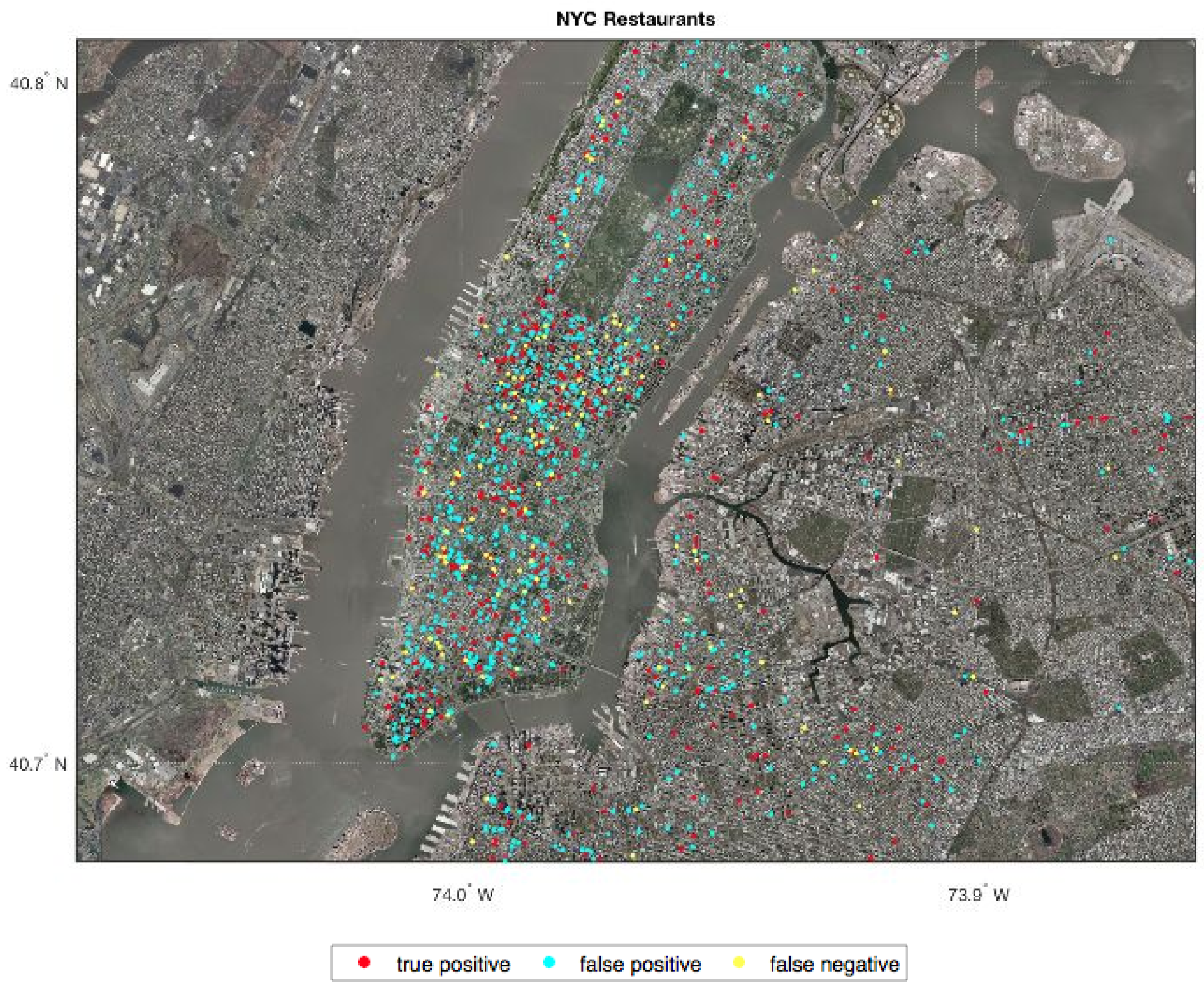

Fig. 10: This map of New York City shows the locations of the restaurants in our test set that our model correctly flagged (true positives), incorrectly flagged (false positives), and failed to flag (false negatives).

References

Datasets

[1] 311 Services. (2017). 311 Service Requests from 2010 to Present. Retrieved from NYC OpenData: https://data.cityofnewyork.us/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9

[2] Department of Consumer Affairs (DCA). (2017). Sidewalk Cafe Licenses and Applications. Retrieved from NYC OpenData: https://data.cityofnewyork.us/Business/Sidewalk-Caf-Licenses-and-Applications/qcdj-rwhu

[3] Department of Health and Mental Hygiene (DOHMH). (2017). DOHMH New York City Restaurant Inspection Results. Retrieved from NYC OpenData: https://data.cityofnewyork.us/Health/DOHMH-New-York-City-Restaurant-Inspection-Results/43nn-pn8j

[4] New York State Liquor Authority. (2017). Liquor Authority Quarterly List of Active Licenses. Retrieved from NY Open Data: https://data.ny.gov/Economic-Development/Liquor-Authority-Quarterly-List-of-Active-Licenses/hrvs-fxs2/

[5] Weather Underground. (2017). Historical Weather. Retrieved from Weather Underground: https://www.wunderground.com/history/

APIs

[6] Google Maps. (2017). Google Maps Geocoding API. Retrieved from https://developers.google.com/maps/documentation/geocoding/

[7] Google Places. (2017). Google Places API. Retrieved from https://developers.google.com/places/

[8] OpenStreetMap. (2017). Nominatim Search Engine. Retrieved from http://nominatim.openstreetmap.org/

Inspiration and Background

[9] City of Chicago. (2016). Food Inspection Forecasting. Retrieved from https://chicago.github.io/food-inspections-evaluation/

[10] Open Data Nation. (2017). Food Inspection Violations Anticipating Risk (FIVAR). Retrieved from http://www.fivar.org/

[11] Department of Health and Mental Hygiene (DOHMH). (2017). *Restaurant Grades* and *Food Service Establishments and the Inspection Cycle*. Retrieved for NYC Health: https://www1.nyc.gov/site/doh/services/restaurant-grades.page

Appendix

New York City Restaurant Inspection Results Variables [3]

Variable Name	**Description**	**Variable Type**
CAMIS	This is an unique identifier for the entity (restaurant)	Plain Text
DBA	This field represents the name (doing business as) of the entity (restaurant)	Plain Text
BORO	Borough in which the entity (restaurant) is located. NOTE: There may be discrepancies between zipcode and listed boro due to differences in an establishment's mailing address and physical location	Plain Text
BUILDING	This field represents the building number for the entity (restaurant)	Plain Text
STREET	This field represents the street name at which the entity (restaurant) is located.	Plain Text
ZIPCODE	zipcode as per the address of the entity (restaurant)	Plain Text
PHONE	Phone Number	Plain Text

CUISINE DESCRIPTION	This field describes the entity (restaurant) cuisine.	Plain Text
INSPECTION DATE	This field represents the date of inspection	Date & Time
ACTION	This field represents the actions that is associated with each restaurant inspection.	Plain Text
VIOLATION CODE	This field represents the violation codes that is associated with each restaurant inspection.	Plain Text
VIOLATION DESCRIPTION	This field is the description that corresponds to the violation codes	Plain Text
CRITICAL FLAG	This indicates if Violation is critical or not.	Plain Text
SCORE	Total Score for a particular inspection. If there was adjudication a judge may reduce the total points for the inspection and this field will have the update amount.	Number
GRADE	• N = Not Yet Graded • A = Grade A • B = Grade B • C = Grade C • Z = Grade Pending • P= Grade Pending issued on re-opening following an initial inspection that resulted in a closure	Plain Text
GRADE DATE	The date when the current grade was issued to the entity (restaurant)	Date & Time
RECORD DATE	The date when the extract was run to produce this dataset	Date & Time

INSPECTION TYPE	The type of inspection. A combination of the program and inspection type.	Plain Text