Forecasting Critical Food Violations in the City of Minneapolis:

An Exploratory Case Study of Open Data

By

Simon Helgeson

A Capstone Project Paper Submitted in Partial Fulfillment of the Requirements for the Degree of  
Master of Science  
In  
Data Science

University of Wisconsin – Eau Claire

Eau Claire, Wisconsin

May 2020

# ABSTRACT

FORECASTING CRITICAL FOOD VIOLATIONS IN THE CITY OF MINNEAPOLIS: AN EXPLORATORY CASE STUDY OF OPEN DATA

Simon Helgeson, Masters in Data Science, University of Wisconsin – Eau Claire, 2020

To prevent the spread of foodborne illness and meet the requirements of Federal and State food regulations, the City of Minneapolis department of public health conducts over 8,000 food safety inspections of restaurants and grocery stores annually (Minneapolis Finance Department, 2019). The goal of this case study is to explore whether it is possible to prioritize food inspections to find the most hazardous food safety violations sooner by assessing risk with a predictive model. This project uses open data sets published by the city of Minneapolis and weather data from the U.S. National Centers for Environmental Information to develop logistic regression and random forest models. The City of Chicago’s open data food inspection risk model provides the foundation of the approach followed in this case study. Two-dimensional kernel density estimation provides a mechanism for aggregating crime and 311 complaints over a rolling time window prior to the date of inspection. This study finds evidence that a model containing data sourced exclusively from data available in Minneapolis’s food inspection data could be employed to find high risk food violations sooner. This study proposes potential future investigation of inspector data and restaurant categories and review from Yelp to further enhance the model.

# TABLE OF CONTENTS

[ABSTRACT ii](#_Toc39444559)

[TABLE OF CONTENTS iii](#_Toc39444560)

[LIST OF FIGURES v](#_Toc39444561)

[CHAPTER 1 - INTRODUCTION 1](#_Toc39444562)

[Project Background 1](#_Toc39444563)

[Problem Definition 2](#_Toc39444564)

[Project Objectives 2](#_Toc39444565)

[Project Inspiration 3](#_Toc39444566)

[CHAPTER 2 – LITERATURE REVIEW 5](#_Toc39444567)

[Forecasting Chicago Restaurant Violations 5](#_Toc39444568)

[Calculating Heat Map Values 7](#_Toc39444569)

[Kernel Density Estimation 7](#_Toc39444570)

[Distance Metrics 8](#_Toc39444571)

[CHAPTER 3 – PRELIMINARY STEPS 10](#_Toc39444572)

[Data Source Selection 10](#_Toc39444573)

[Minneapolis Open Data 10](#_Toc39444574)

[NOAA Weather 11](#_Toc39444575)

[Data Validation and Cleansing 11](#_Toc39444576)

[CHAPTER 4 – METHODOLOGY 14](#_Toc39444577)

[Feature Creation 14](#_Toc39444578)

[Heat Values – Kernel Density Estimation 14](#_Toc39444579)

[Inspection History 15](#_Toc39444580)

[Dummy Variables 15](#_Toc39444581)

[Model Development and Training 16](#_Toc39444582)

[Model Tuning 18](#_Toc39444583)

[CHAPTER 5 – RESULTS AND DISCUSSION 19](#_Toc39444584)

[Variable Importance 19](#_Toc39444585)

[Prediction Assessment 20](#_Toc39444586)

[CHAPTER 6 – CONCLUSION AND RECOMMENDATIONS 22](#_Toc39444587)

[Conclusion 22](#_Toc39444588)

[Recommendations 22](#_Toc39444589)

[REFERENCES 24](#_Toc39444590)

# LIST OF FIGURES

[Figure 1: Confirmed Foodborne Outbreaks (Minnesota Department of Health Infectious Disease Epidemiology, Prevention and Control Division, 2014) 1](#_Toc39447292)

[Figure 2: City of Chicago diagram showing model development goal of changing inspection schedule to find critical health violations sooner (Schenck, et al., 2015) 6](#_Toc39447293)

[Figure 3 Feature importance for best logistic regression model 20](#_Toc39447294)

[Figure 4 Feature importance for best random forest model 21](#_Toc39447295)

[Figure 5 Classification report for holdout test data 21](#_Toc39447296)

# CHAPTER 1 - INTRODUCTION

## Project Background

Foodborne illness is something that affects most humans at some point in their life. Many of us can remember feeling ill after eating food at a restaurant with less than sanitary kitchen conditions. The news in recent years has been filled with reports of E. coli and norovirus outbreaks from popular restaurant chains like Chipotle. Chipotle’s stock and customer base declined after large outbreaks of food poisoning at the chain. (Yaffe-bellany, 2019)

A 2014 report published by the Minnesota Department of Health confirmed foodborne illness outbreaks have risen in recent years. (Minnesota Department of Health Infectious Disease Epidemiology, Prevention and Control Division, 2014). Whether this is due to better reporting or an actual rise in prevalence, it has become clear that foodborne illness is an issue that affects many people every year.

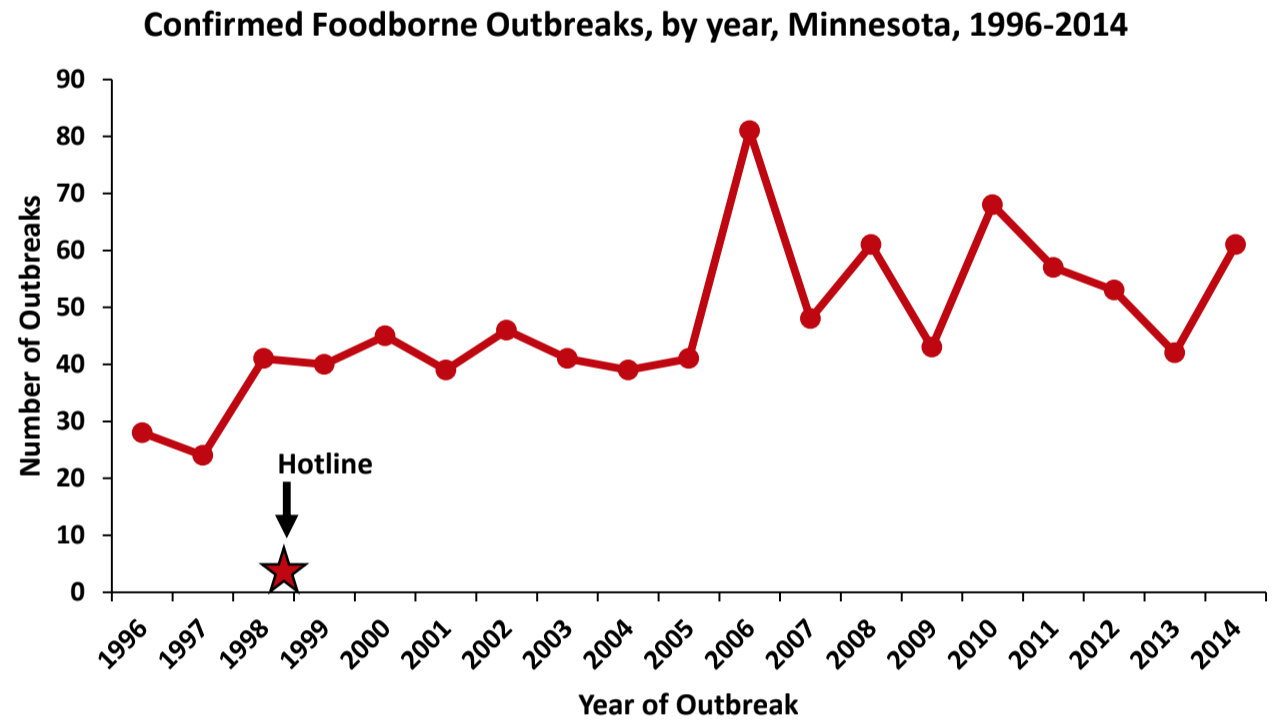


Figure : Confirmed Foodborne Outbreaks (Minnesota Department of Health Infectious Disease Epidemiology, Prevention and Control Division, 2014)

A primary avenue for preventing or reducing the incidence of foodborne illness is through routine health inspections conducted by city government. Inspections can unearth unsanitary conditions and potential health risks at restaurants and food stores. The city of Minneapolis conducts over 8,000 food safety inspections per year (2020 operating budget). Historically city food inspections have been mostly routine or complaint driven. The city is moving to a more proactive inspection program that focuses on education and consultation with businesses, reducing the use of traditional enforcement mechanisms like tickets and violations. (Minneapolis Finance Department, 2019)

## Problem Definition

Despite these changes, the city faces an increasing threat of foodborne illness and a limited budget for appropriate staffing. According to the 2019 operating budget, Minneapolis’s hospitality industry has experienced a 58% increase in sales in eight years. The city dealt with a record 15 food-related disease outbreaks in 2018 and booked 1000 hours of staff overtime. (Minneapolis Finance Department, 2019) These statistics point to potential strategic and financial value in improving the prioritization and scheduling of these inspections.

## Project Objectives

1. Create a modeling dataset that merges Minneapolis food inspection data with crime, 311, and weather data.
2. Establish a predictive risk score for each business prior to the inspection date. Use risk scores to prioritize food safety inspections for high risk businesses.
3. Share results with staff at the city of Minneapolis and promote publishing of open data sets by government agencies.
4. Explore potential for integrating restaurant category and review data from Yelp/Google.

## Project Inspiration

My work at Allstate provided me with opportunities to attend Analytics Lunch and Learn events where analytics professionals from across the company would present on their projects. A few years ago, I attended a presentation put together as part of Allstate’s project lightbulb program. Allstate’s project lightbulb allowed high performing employees the opportunity to spend a percentage of their work hours on passion projects that were not necessarily tied to their team’s objectives.

As part of this program, Allstate data scientists established a partnership with the City of Chicago to see if they could help improve city services through their analytical expertise. At the time, Chicago had a booming restaurant industry and a budget for food safety inspections that limited the available staff. Chicago was looking for a way to prioritize inspections to target restaurants that were more likely to have a critical food safety violation.

As I was exploring options for a capstone final project, I discovered that the city of Minneapolis had recently released an open government dataset for historical food inspections. I immediately thought of the project Allstate conducted with City of Chicago and decided that replicating the study they did with data from Minneapolis would offer a great learning opportunity and, if successful, the chance to serve my local community. Once I began to build the project plan, I discovered several additional benefits of this project.

* opportunity to review code assembled by other data scientists
* opportunity to translate code written in the R language to Python
  + Python is the primary language I use at work
  + Most of Arity’s data science models are written in Python
* opportunity to collaborate with government and promote open data sets
  + interested in potential future work in the public service sector
* opportunity to work with geospatial data
  + geospatial analysis is critical to many careers in the environmental field
  + geospatial analysis is critical to my work at Arity
* opportunity to learn about the field of environmental health

These educational opportunities plus the potential to develop something of value for the City of Minneapolis convinced me that this was a project would be a worthy culmination of time invested earning my masters in Data Science.

# CHAPTER 2 – LITERATURE REVIEW

## Forecasting Chicago Restaurant Violations

In 2014, the City of Chicago and Allstate developed an innovative partnership to explore the potential for using a model to help prioritize food inspection scheduling for licensed food service providers. The successful results of this partnership and subsequent private citizen contributions to the project have inspired several other cities to test the methodology. A recent publication in the journal *Government Information Quarterly* highlighted Chicago’s project as a great example of open government data (OGD)-driven public service co-creation. The authors argue that the keys to the success of the project were “motivated stakeholders, innovative leaders, proper communications, and existing OGD portal, external funding, and agile development.” (McBride, Aavik, Toots, Kalvet, & Krimmer, 2019) My analysis is rooted in the approach and conclusions generated from this study. An overview of this study serves as groundwork for laying out my methodology.

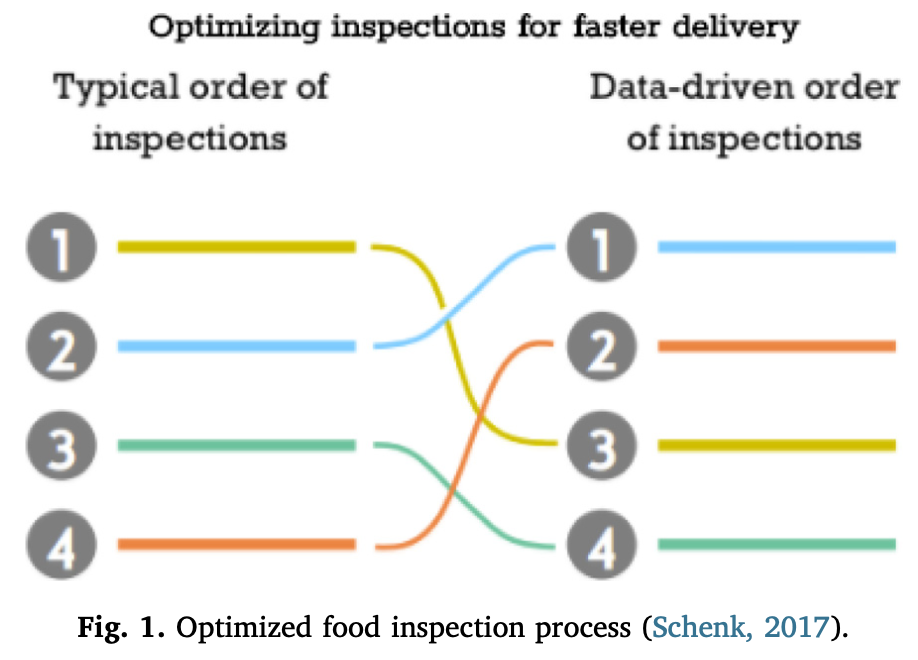
A 2015 publication entitled “Forecasting Critical Restaurant Violations in Chicago” describes the predictive model that the City of Chicago built to “identify the presence of a critical violation in a particular food establishment.” The goal of their machine learning model was to “identify the riskiest restaurants earlier, thereby reducing the length of exposure of risky restaurants to patrons” Inspectors were able to order routine inspections so that unsafe restaurants were inspected first, finding and helping business rectify the most hazardous health risks an average of 7 days earlier than inspectors who were not using the model to prioritize inspections. (Schenck, et al., 2015)

Figure : City of Chicago diagram showing model development goal of changing inspection schedule to find critical health violations sooner (Schenck, et al., 2015)

The authors of the model tested a wide variety of variables for inclusion in the model, ultimately discovering that the most predictive feature was the name of the inspector conducting the investigation. To protect the privacy of the names of the inspectors, inspectors were clustered based on likelihood to find a critical violation. In addition, Chicago ultimately included the following variables in the predictive model:

1. Establishments that had previous critical or serious violations
2. Three-day average high temperature.
3. Nearby garbage and sanitation complaints.
4. The type of facility being inspected.
5. Nearby burglaries.
6. Whether the establishment has a tobacco license or has an incidental alcohol consumption license.
7. Length of time since last inspection.
8. The length of time the establishment has been operating.

(McBride, Aavik, Toots, Kalvet, & Krimmer, 2019)

## Calculating Heat Map Values

To better understand how heat map values factor into a model that is potentially sensitive to both geospatial and time elements, I found an article describing the use of spatio-temporal kernel density estimates for crime hotspot mapping helpful. According to the authors, there are two primary methods for assessing geospatial correlation with a target variable (crime).

The first option is to aggregate incidents to counts and calculate rates based on geographic boundaries such as census units. Once these aggregate counts are developed, multivariate regression analysis may reveal relationships between the target variable and a range of other variables such as socio-economic conditions, neighborhood demographics and land use types. (Hu, Wang, Guin, & Zhu, 2018)

## Kernel Density Estimation

The second option is to generate predictive hotspot mapping by identifying spatial clustering patterns of incidents. Kernel density estimation (KDE), a popular hotspot mapping approach, “converts point incidents to a density surface that summarizes the point distribution.” KDE “places a kernel over a predefined area around the location,” assigns “more weights to nearby events than distant ones,” and sums the weighted events within the kernel. “Areas on the surface with high density values above a pre-defined threshold are defined as hotspots.” (Hu, Wang, Guin, & Zhu, 2018)

Two-dimensional KDE without a predefined hotspot density threshold is a popular method for producing geographic heatmap data visualizations. The city of Chicago food inspection project used KDE to produce heat values for crime, sanitation, and garbage cart requests. To account for changes over time in the location and density of clusters, a time-boxed window for predicting heat (density) is useful. The reality of cities is that conditions like crime change over time. Crime risks may be temporarily elevated in certain geographic regions, as neighborhood socioeconomic conditions change or when policing pushes crime to different areas. (Hu, Wang, Guin, & Zhu, 2018)

KDE is sensitive to search bandwidth, a parameter that controls the smoothing and width of clusters. A larger bandwidth returns “smoother and bigger event clusters,” whereas a smaller bandwidth detects “small, spiky event hotspots.” Selecting the optimal bandwidth is critical to effectively detecting event clusters. (Pedegrosa, 2020)

In addition to the bandwidth, practitioners using KDE also need to select an appropriate kernel and a distance metric. The simplest and most commonly used kernel is the gaussian normal kernel. It estimates probability densities by scaling a normal distribution (mean 0, standard deviation 1) by the actual latitude and longitude. In essence a Gaussian distribution is fit to each point with weights. Points farther from the center have less weight than points near the center. (VanderPlas, 2016)

## 

## Distance Metrics

Euclidean distance (straight line distance in 2D plane) is often selected as the method for estimating kernel density. When working with geodetic Earth coordinates like latitude and longitude, event density estimation should consider using a distance calculation that takes into account the curvature of the Earth such as Haversine distance. However, when considering a small geographic region such as a city, the differences between haversine (spherical) distance and Euclidean distance are negligible. (Agafonkin, 2016). In situations where the events are human activities, realistically the distance between events is measured over a grid like network of streets. In these situations, it may be more practical to use a distance measure like network distance (Hu, Wang, Guin, & Zhu, 2018), which would consider the time it takes to travel between points using the shortest route (something similar to the recommendations of a GPS navigation system or Google Maps).

# CHAPTER 3 – PRELIMINARY STEPS

## Data Source Selection

### Minneapolis Open Data

Most of the data incorporated into the project can be found on the Minneapolis public data portal (opendata.minneapolis.gov). The three primary Minneapolis city data sets are:

* food inspections (<http://opendata.minneapolismn.gov/datasets/food-inspections>)
* police incidents (<http://opendata.minneapolismn.gov/datasets/police-incidents-2019>)
* public 311 (<http://opendata.minneapolismn.gov/datasets/public-311-2019-ytd>)

The open data portal has police incidents and 311 data available for years 2011-2020. Unfortunately, the open data portal only stores inspections data from 2017-2020. In order to have a more robust sample size for model development, I asked the Minneapolis inspections team to supply additional years of inspections data dating back to 2014. Because most restaurants are only inspected once or twice per year, it seemed important to have more than 3-6 inspection scores available for each restaurant to develop an accurate prediction for each business going forward.

After interviewing Stephen Collins, an Allstate data scientist who contributed to the City of Chicago modeling project, I learned that the one of the most powerful predictors of whether a critical food safety violation would be found was the person doing the inspection. Some inspectors tend to be more critical while others are more lenient. Based on this learning, I asked the Minneapolis health inspections team to add this feature to the expanded set of inspections data.

Fortunately, the inspections team agreed to help me out by providing the extended history and the name of the inspector. They delivered an excel file with inspection data dating back to 2008. This extended data provides a great history of results for training models.

### NOAA Weather

I requested historical weather data from NOAA Climate Data Online for meteorological readings from the Minneapolis St. Paul International Airport dating back to 2011. (US Department of Commerce, 2020) The City of Chicago uses the forecast.io API for daily temperature maximums as part of their model. (Schenck, et al., 2015) Unfortunately, Dark Sky, the company powering the forecast.io API, is no longer accepting new applications due to a recent merger with Apple. (Grossman, 2020) The data set from NOAA includes the daily average drybulb temperature and daily maximum drybulb temperature.

## Data Validation and Cleansing

Minneapolis offers multiple ways of interacting with the open data portal. In addition to traditional download formats like a spreadsheet, Minneapolis offers two API formats for their data: a proprietary ARCGIS format and an open source GeoJSON format. GeoJSON data format allows geographic information like polygon shapes and coordinates to be encoded in a nested structure similar to traditional JavaScript Object Notation (Butler, et al., 2016)

GeoJSON format is readily processed by the GeoPandas package in python. GeoPandas is a useful tool for developing maps that aggregate geopoints within a polygon-shaped spatial region, such as a city neighborhood. (McBride, et al., 2020) I developed a python script to read the various APIs and load the data directly into Pandas data structures. Unfortunately, Minneapolis stores each year of historical data in a separate API URL, necessitating a manual lookup of each year’s API path.

To gather all the data, I merged inputs from multiple years into a single file containing all the available years of data. An important step I took in merging the data sets was cross-referencing field names across the years. In some cases, field names changed from one year to the next even though the underlying data remaining the same. I had to synchronize the data across the years to obtain a single column for each of the relevant fields.

The crime data set format changed after the city migrated from its 28-year old CAPRS (Computer Assisted Police Records System) to a management system called PIMS (Police Incident Management System) in June 2018 (Libor, 2018). Fortunately, the relevant fields – the type of crime, data of the crime report, and geographic latitude and longitude existed in both systems and appeared to be well populated for nearly all records. Police incidents include a variety of the crime types. To align with the City of Chicago’s methodology, I filtered the Minneapolis crime to only include residential and business burglaries.

The City of Chicago’s model benefited from a dataset of 311 incidents separated based on the type of complaint or request. Minneapolis aggregates all 311 incidents into a single annual data set. To classify the data, Minneapolis attaches a reason name and type name to each incident. While the incident reason is a broad categorization, the incident type provides a finer level of detail. I selected incident types “Solid Waste – Overflowing Litter” and “Commercial Food Safety/ Sanitation Complaint” for further analysis due to their potential link to food safety. Overflowing litter might attract rats and food safety complaints might be tied to safety issues that could be found in an inspection.

The weather data from NOAA was missing daily observations for 3 days. I used linear interpolation to fill in the missing readings with the midpoint between the readings for one day before and one day after the missing measurement.

# CHAPTER 4 – METHODOLOGY

## Feature Creation

### Heat Values – Kernel Density Estimation

One of the biggest challenges in this project was to develop a way to link the crime and 311 data to the inspections data. The city of Chicago model uses kernel density estimation to generate heat values for the latitude/longitude of the inspection based on the density of crime/sanitation requests in a timeframe leading up to the date of the inspection. I decided to look at a 90-day window of crime/sanitation events leading up to the data of the inspection and create a 2-dimensional kernel density estimate for the latitude/longitude of the inspection.

I decided to conduct all of my feature creation and modeling in Python. To my knowledge, Python does not have a KDE implementation that mirrors the one available in R’s MASS package. As an alternative, I used the Scikit-learn implementation of KDE to prepare heatmap features. Scikit-learn has a built-in cross validation tool (GridSearchCV) for selecting an appropriate bandwidth. To get good separation of event clusters, I iterated through the inspections data set, testing a range of bandwidth values between 0.001 and .1 for both the crime and 311 data sets. Based on the mean of the reported results, I selected a bandwidth of 0.004 for crime and 0.009 for 311 data.

I applied these bandwidth values to fit a kernel density model to geopoints in the 90-day trailing window preceding the inspection date. I then inputted the latitude and longitude values of the inspection into the kernel density model to produce a probability density estimate. The probability density serves as the heat value of nearby crime and 311 estimates. Heat values serve as input variables for predicting critical (priority 1) violations.

### Inspection History

In addition to the heat values, I created features from the food inspections data set to mirror some of the variables used in Chicago’s model. For each business I looked backwards in time through the inspections data to find the first inspection date for that business to calculate the time difference between their first inspection and the current inspection. With 9 years of inspection data, this serves as a proxy for the amount of time the business has been in operation. Businesses that have been operating longer have had multiple food safety inspections and might behave differently from a newer restaurant. I created an indicator variable to identify the first inspection for each business.

Based on the theory that businesses with historical violations might continue to violate the health code, I created variables that look back to the inspection 1 period prior to the current inspection and aggregate the health code violations based on priority level. This process results in three additional columns of data, one for each priority level.

### Weather

Mirroring the City of Chicago, the feature I created for weather was 3 day moving average maximum temperature for the 3 days leading up to the date of the inspection. This feature is based on the idea that weather systems tends to last a few days and have similar temperatures. Weather could factor into food safety if heat causes coolers to break down or food to spoil faster.

### Dummy Variables

To support the inclusion of categorical variables in my models, I transformed the categorical variables into dummy variables. Dummy variables are binary variables that replace categories. For example, if there are three values in the original categorical variable, the process of creating dummy variables will replace the single column with three binary columns, each indicating the presence or absence of one of the categories. To avoid multicollinearity, one of the factors levels should be dropped in the dummy creation process. The dropped level will serve as the default baseline comparison.

The facility category variable provides a good example of the dummy variable creation process. The cleansed and filtered data contains both restaurant and grocery facilities. The dummy variable created for facility category is an indicator variable identifying whether or not the facility is a restaurant. If the value of the dummy variable is 1 the business is a restaurant, but if the value is zero the facility is a grocery store.

## Model Development and Training

The first step in preparing the data set for modeling was splitting the data into train, validation, and test groups. I wanted to hold out a portion of the data to assess the feasibility of using the model to score the inspections, so I set aside data from the last 3 quarters of 2019 (April-December) to use in testing my final model. I randomly selected 80% of the remaining inspections as the training data set and 20% of the remaining observation as the validation set. Each of the models was fit using the training data set and then assessed and tuned with the validation set.

I tested two modeling approaches for predicting the likelihood that an inspection would result in at least one priority 1 violation: logistic regression and random forest. The final set of features I used for training included the following:

1. *first\_inspection\_ind*: indicator variable for whether or not this was the first recorded inspection for the business
2. *years\_since\_last\_inspect*: number of years (days/365) since the last inspection
3. *prior\_priority\_*1: number of priority 1 violations recorded in the previous inspection
4. *prior\_priority\_2*: number of priority 1 violations recorded in the previous inspection
5. *prior\_priority\_3*: number of priority 3 violations recorded in the previous inspection
6. *years\_since\_first\_inspection*: number of years (days/365) since the first recorded inspection for the business
7. *burglary\_heat*: KDE heat value for 90-day window of nearby burglary events
8. *sanitation\_heat*: KDE heat value for 90-day window of nearby sanitation complaints
9. *threeDayAvgMaxWeather*: rolling three day average of the daily maximum weather for the 3 days leading up to the inspection date
10. *risk\_1*: binary indicatory of whether or not the business is categorized as risk 1 (highest risk) based on business class. Risk 1 facilities are typically full-service restaurants.
11. *risk\_2*: binary indicator of whether or not the business is categorized as risk 2 (medium risk) based on business class. Risk 2 restaurants do not prepare large amounts of food in advance.
12. *restaurant\_ind*: binary indicator of whether or not the business is a restaurant; if 1, restaurant, else grocery store.

## Model Tuning

After fitting the initial model, I used recursive feature elimination to remove variables the were not contributing to predictive accuracy. Recursive feature elimination is implemented in Scikit-learn and can be applied to the various modeling approaches available in Scikit-learn. (Shetye, 2019)

When I ran recursive feature elimination on the logistic regression model, the heat variables for sanitation was removed from the model, leaving 10 other variables behind. The resulting predictive accuracy was 70.7% I tried going back and adjusting the heat values to cutoff outliers similar to the way the City of Chicago handled the data in their model, but this approach resulted in the removal of both sanitation and burglary heat variables in recursive feature elimination. It appears that, despite the considerable effort invested in developing these features, they are not useful in creating predictions.

Recursive feature elimination on the random forest model resulted in a similar removal of heat features. The 3-day average max temperature feature was also removed. Only 7 of the original 12 variables were left behind ('years\_since\_last\_inspect', 'prior\_priority\_1', 'prior\_priority\_2', 'prior\_priority\_3', 'risk\_1', 'risk\_2', 'restaurant\_ind'). With only these variables, random forest achieved a slightly better predictive accuracy than the logistic regression model (71.1%). All of these variables come from food inspections data set and require no external data.

# CHAPTER 5 – RESULTS AND DISCUSSION

## Variable Importance

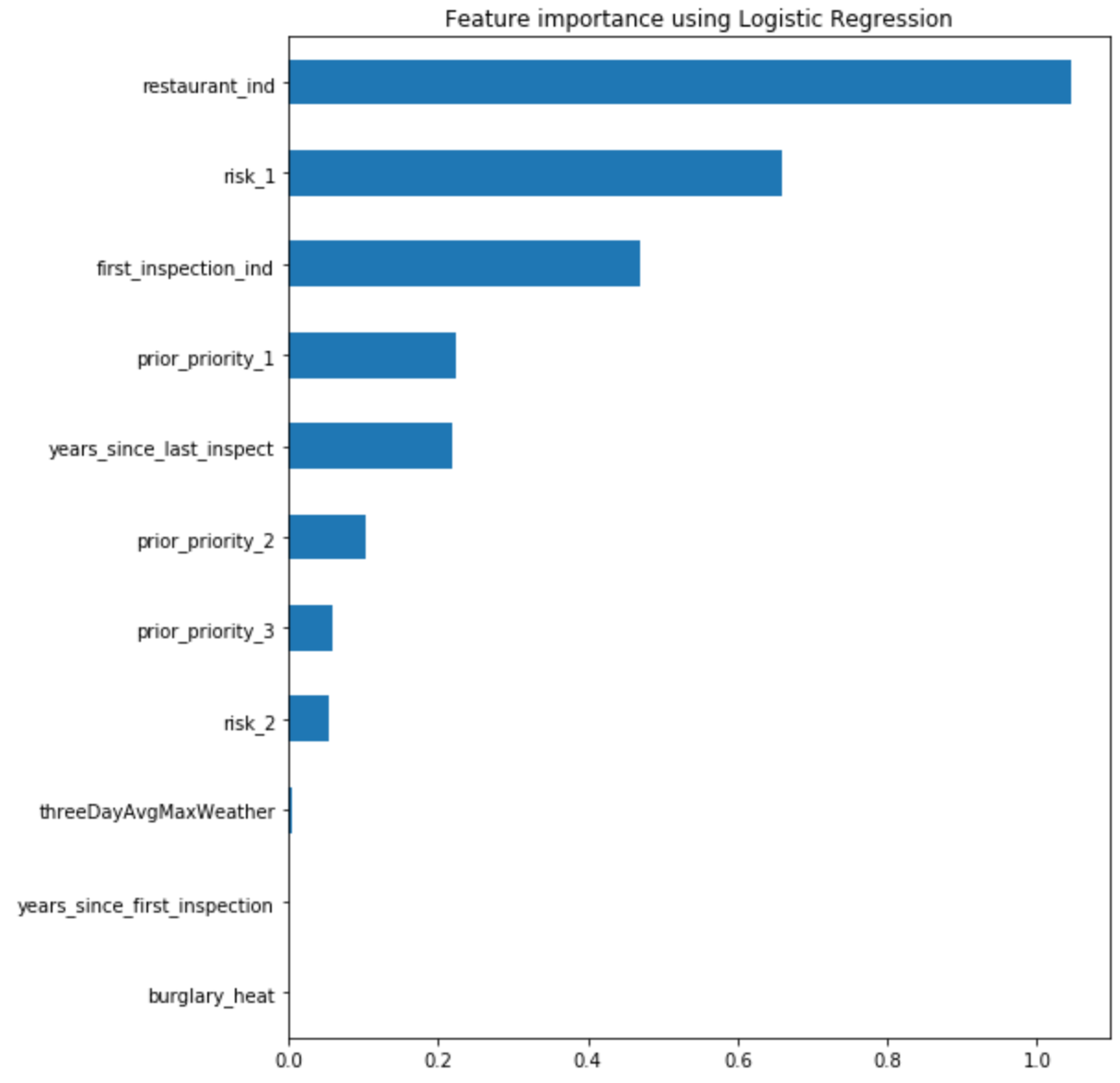
Based on the best models selected through recursive feature elimination, the most important features for the logistic regression model are whether the business is a restaurant or grocery store (restaurant\_ind), risk\_1, first\_inspection\_ind, prior\_priority\_1, years\_since\_last\_inspect, prior\_priority2, prior\_priority\_3, and risk\_2. Weather and burglary heat appear to provide almost zero benefit to the prediction results. 

Figure Feature importance for best logistic regression model

Feature importance is similar for random forest. The top two features align with the logistic regression model. With the exception of first\_inspection\_ind, which is not included in the random forest model, the other variables appear to have similar rankings of feature importance.

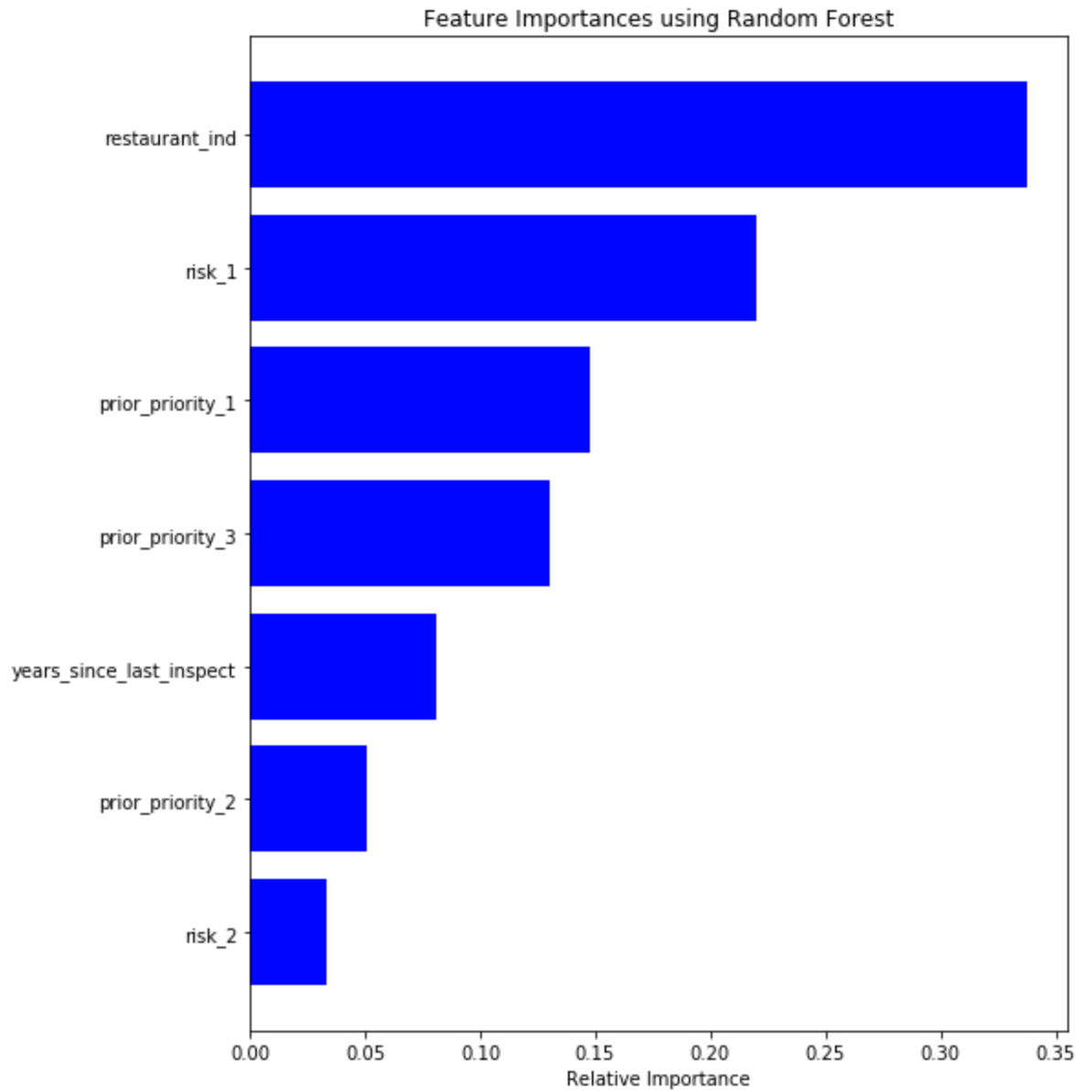


Figure Feature importance for best random forest model

## Prediction Assessment

After recursive feature elimination, the remaining features were fit to the test data set to obtain final scores. The random forest model with 7 features achieved a 68% accuracy on the holdout test data. The model correctly identified 722 inspections with a critical violation and falsely predicted a priority 1 violation for 364 inspections that did not have violation. The full classification report follows:

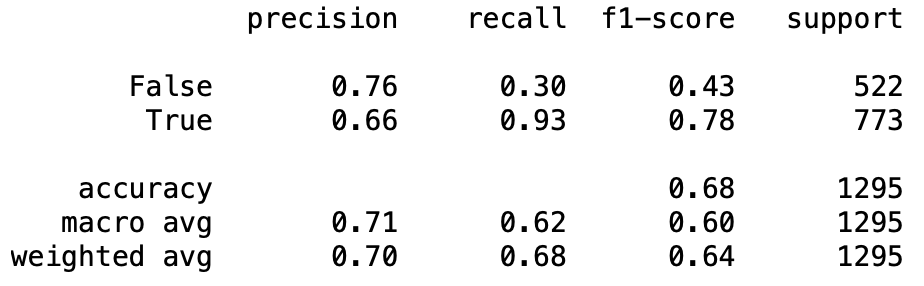


Figure Classification report for holdout test data

To examine how much of a time savings could be obtained in finding critical inspections if inspections in the 9-month period were reordered, I took the original list of inspection IDs and reprioritized them according to predicted probability score so that the earliest inspections in the period were the inspections with the greatest scores. I then took the original list of inspection dates and assigned them in consecutive order according to the businesses with the highest risk scores. This results in a number of inspections on each calendar day that matches the original data set. By reprioritizing in this way, priority 1 violations were found an average of 27 days earlier or a median of 31 days earlier.

Presenting a prioritized list of inspections to the city may be a good way of sharing these results. According to the article published by McBride et. al discussing the results of the Chicago food inspections modeling project, the data scientist’s working on the project had ambitious ideas for how to present the statistical reasoning behind their model in a tool. It turned out that the inspection team preferred a simple list of prioritized inspections. (McBride, Aavik, Toots, Kalvet, & Krimmer, 2019)

Unfortunately, it may not be realistic to use a 9-month data set to perform these prioritizations. State and federal food safety laws limit of flexibility of how often businesses are inspected. If Minneapolis inspects a business earlier than regularly scheduled because the model predicts critical violations, it could be a burden on the business if the model is wrong. Businesses might complain to city council and pressure the inspections team to stick to regularly scheduled routine visits. Revisiting the evaluation with a 1-month dataset and exploring how much of a difference the model can make when there is only 30 days of routine inspections to reprioritize would be a more realistic evaluation.

# CHAPTER 6 – CONCLUSION AND RECOMMENDATIONS

## Conclusion

If Minneapolis would like to use a model to schedule inspections, it may make sense to start out with a very simple model. Using just the variables available in the food inspections data set, I was able to achieve a meaningful improvement in the time it takes to find critical violations on a 9-month evaluation data set. If Minneapolis is able to reorder inspections that were planned for a single month, they could find critical violations sooner. Further work is needed to assess other modeling approaches to make the integration of external data sets helpful in developing predictions.

## Recommendations

To enhance the predictive ability of the model, further investigation could focus on leveraging the inspector dataset. I had difficulty determining how to make good use of the data. One option would be to grouping inspections by inspector and aggregating inspection scores to see which inspectors tend to find more critical violations per inspection compared to peers. In my initial investigation I found that, after filtering the data to only inspectors with a history of at 200 inspections, some inspectors found critical violations in over 70% of their inspections while other inspectors found violations less than 50% of the time. Properly assessing this data would require deeper partnership with the city. The City of Chicago implemented a penalty for using inspector data in their models so that they could account for this effect in their results (Schenck, et al., 2015). Taking a similar approach with the Minneapolis data would likely enhance the accuracy of the model.

Gathering and integrating Yelp or Google Places data could be another avenue for enhancing the data set. The Yelp Fusion business search API allows a developer to pass through data from another source (i.e. Minneapolis food inspections) and returns a business ID match. This business ID can be passed through to business ID and review endpoints to gather matching data including the restaurant category (i.e. American, French, wine bar etc.), recent reviews, and more. Gathering Yelp API matches proved to be beyond the scope of my investigation but it could provide data points that would help boost the predictive ability of the model. (Yelp, 2020)

# REFERENCES

Agafonkin, V. (2016, May 2). *Fast geodesic approximations with Cheap Ruler*. Retrieved from Medium: https://blog.mapbox.com/fast-geodesic-approximations-with-cheap-ruler-106f229ad016

Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S., & Schaub, S. (2016, August). *The GeoJSON format.* Retrieved from Internet Engineering Task Force: https://tools.ietf.org/html/rfc7946

City of Chicago. (2017). *Food Inspection Forecasting. Optimizing Inspections with Analytics.* . Retrieved from https://chicago.github.io/food-inspections-evaluation/

City of Minneapolis. (2020, April 28). *Food Inspections*. Retrieved from Open Data Minneapolis: https://opendata.minneapolismn.gov/datasets/food-inspections

Grossman, A. (2020, March 31). *Dark Sky Has A New Home*. Retrieved from Dark Sky: https://blog.darksky.net/

Hu, Y., Wang, F., Guin, C., & Zhu, H. (2018). A spatio-temporal kernel density framework for predictive crime hotspot mapping and evaluation. *Applied Geography, 99*, 89-97.

Libor, J. (2018, June 13). *Minneapolis Police Department unveils new records system to help it keep up with modern crime.* Retrieved from Star Tribune: https://www.startribune.com/minneapolis-police-department-unveils-new-records-system-to-help-it-keep-up-with-modern-crime/485452391/

McBride, J., Van den Bossche, J., Wasserman, J., Jordahl, K., John Wolf, L., Fleischmann, M., & Rocklin, M. (2020, February 17). *Geopandas*. Retrieved from Github: https://github.com/geopandas/geopandas

McBride, K., Aavik, G., Toots, M., Kalvet, T., & Krimmer, R. (2019). How does open government data driven co-creation occur? Six factors and a ‘perfect storm’; insights from Chicagos food inspection forecasting model. *Government Information Quarterly, 36*(1), 88-97.

Minneapolis Finance Department. (2019, December 12). *Minneapolis 2020 Budget.* Retrieved from City of Minneapolis: http://www.minneapolismn.gov/budget/2020-budget

Minneapolis, C. o. (2018, June 29). *Police Incidents 2018.* Retrieved from Minneapolis Open Data: https://opendata.minneapolismn.gov/datasets/police-incidents-2018

Minnesota Department of Health Infectious Disease Epidemiology, Prevention and Control Division. (2014). *Outbreak Statistics: Summary of Gastroenteritis Outbreaks in Minnesota.* Retrieved from Minnesota Department of Health: https://www.health.state.mn.us/diseases/foodborne/outbreak/outbreaksummary.html

Pedegrosa, F. (2020, March). *Density Estimation*. Retrieved from Scikit-learn: Machine Learning in Python: https://scikit-learn.org/stable/modules/density.html

Ripley, B., Venables, B., Hornik, K., Gebhardt, A., & Firth, D. (2020, April 26). *Two-Dimensional Kernel Density Estimation.* Retrieved from ETH Zurich Department of Mathematics: https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/kde2d.html

Schenck, T., Leynes, G., Solanki, A., Collins, S., Smart, G., Albright, B., & Crippin, D. (2015, May 15). *Forecasting Resturants with Critical Violations in Chicago.* Retrieved from Github: https://github.com/Chicago/food-inspections-evaluation/

Shetye, A. (2019, February 11). *Feature Selection with sklearn and Pandas*. Retrieved from Medium: https://towardsdatascience.com/feature-selection-with-pandas-e3690ad8504b

US Department of Commerce. (2020, April 28). *Climate Data Online Data Tools*. Retrieved from NOAA: National Centers for Environmental Information: https://www.ncdc.noaa.gov/cdo-web/datatools

VanderPlas, J. (2016). *Python Data Science Handbook: Essential Tools for Working with Data.* Sebastopol: O'Reilly Media.

Yaffe-bellany, D. (2019, July 23). *Chipotle, With Food Safety Issues Behind It, Recovers Strongly.* Retrieved from New York Times: https://www.nytimes.com/2019/07/23/business/chipotle-stock-earnings.html

Yelp. (2020, May 3). *Fusion API business search*. Retrieved from Yelp: https://www.yelp.com/developers/documentation/v3/business\_search