PREDICTING FOOD INSPECTION RESULTS IN CHICAGO

Ziqin Wang & Mingfang Zhang

MUSA 507 FINAL PROJECT

Introduction

The Department of Public Health in Chicago is responsible for promoting public health in the areas of food safety and sanitation to prevent the spread of foodborne disease. While Centers for Disease Control and Prevention estimated that there are around 3000 people die from foodborne diseases each year in the United States. In order to protect Chicagoans from this potentially threat, City of Chicago's Department of Public Health completes more than 20,000 inspections on more than 16,000 food establishments every year (shown as *Figure 1*). Based on these food inspections as well as some collected data, we could build a predictive model and show the results of food inspections and predictive analysis more friendly and concisely with a Chicago Smart Food application.

By searching information in our Chicago Food rate application, customers could check whether the food establishment they are are going is safe or not, they could also provide feedbacks for these food establishments. For inspectors, this application could provide them previous inspection results, in this app they could create a list of high-risk establishments and complete the inspection report electronically. Also, supervisors in the Department of Public Health in Chicago could know the distribution of failed food establishments and mange the progress of each inspector in time, with all these inspection data they could also educate food business and address food related emergencies. This report is written to introduce concepts and procedures behind the Chicago Food Rate application, especially focus on the how to develop and test the core logistic regression model applied in this project.

Hypotheses

Before taking steps to build the predictive model on food inspections, we need to figure out a problem, "When would food establishments be more likely to be inspected as fail?" To answer this question, we put forward the following hypotheses:

1. Establishments with "bad" history would be more likely to fail in food inspection.
2. Establishments would be more likely to fail a food inspection in spring and summer months in a year.
3. The more frequent it used to be inspected, the more likely an establishment would be to fail in food inspection.
4. Establishments are more likely to fail a food inspection in some disorganized neighborhoods.
5. Establishments are more likely to fail a food inspection if they located in high crime neighborhoods.
6. Some certain type of establishments, such as restaurant and grocery store compared to school or day care center, are more likely to fail a food inspection.
7. The longer it has been the previous inspection, the more likely an establishment would be to fail an inspection.

With all these hypotheses, we are ready to start our journey of building food inspection prediction model and designing pragmatic Chicago Smart Food application.

Data

When looking through variables in food inspection, we found we could build predictors, such as inspection month of the year and establishments types, directly from this dataset. We could also use the establishments' previous inspections results as independent variable, for example, if we use food inspections in 2015 to build logistic regression, the inspection results and serious violations (violation #1-29) in 2014 could be served as predictors. Predictors such as the length of days from previous inspection and total number of inspections received in the previous calendar year could also be included as predictors.

Besides food inspection data, we also found some interesting data from the City of Chicago's Open Data Portal, such as crime and 311 service data (sanitation complains and garbage carts) in 2015. The full list of variables and descriptions are provided in *Table1*.

Methodology

After establishing our hypotheses and obtaining all needed data, we start to deal with data in ArcGIS. We intended to include as much probable variables as possible and determine whether they are statistically significant predictors later when we really run the model. Predictors such as inspection results from last year, length of time from the previous inspection, whether received serious violations during the previous inspection and total number of inspections received in the previous calendar year could be created in ArcGIS by matching inspections by establishment license number. Crime data are spatial joined by calculating the number of crimes within 300 feet area around the food establishment. The sanitation complains and garbage carts (311 data) are spatial joined by sum up the number of sanitation complains or garbage carts within 200 feet area around the establishment. After all these joins, we create a total of 22 independent variables to predict fail/pass food inspection outcome.

We did procedures stated above for both food inspections in 2015 and food inspections in 2016. Food inspection in 2015, would be served as training data. In order to prohibit malposition when matching the inspection results in 2014 by license number, we only keep the first inspection in each calendar year for the same establishment, which means there are no establishment included more than once in the dataset. After data cleaning, there are 11101 valid food inspection records in 2015. Food inspection in 2016, would be served as testing data. After removing duplicate, there are around 9858 valid records to be tested in food inspection 2016.

Using the training data (2015), we run cross-tabulation to test whether there is serious multicollinearity between predictors, then wen run stepwise regression in R to identify the most statistically significant predictors of an inspection's fail. The predictors are narrowed from 22 to only 10. We also select 3 separate test set neighborhoods in 2015, which are Chinatown, Hyedepark and Woodlawn, to test whether our model works better for certain types of neighborhoods. After building logistic regression model with these 10 independent variables, we did a bunch of goodness-of-fit measures for both training data and testing data to assess the quality of the model. (All procedures are shown in *Figure 2*)

Results

After running the final regression model, we found that there are only most of variables we expected to be relevant with dependent variable: failed inspection. While sanitation complains, type of establishment and length of time from the precious inspection were eliminated from the model, inspection history, garbage carts, previous serious violations, number of inspection times in previous year, crime and neighborhood are proved to be valuable predictors. All predictors used in the final logistic regression model are listed in *Table 2.*

As those goodness-of-fit indicators shown in the *Table 3*, the model got positive results. The McFadden's pseudo-R^2 is 0.323, in the 0.2 to 0.4 range that is usually used to be described as well-fitted models. The Received Operating Characteristic (ROC) Curves are shown in *Figure 3*, and the Area under the Curve (AUC) was 0.9098 for the training data and 0.8974 for the testing data, the accuracy of this model could be described as good.

Instead of predicting a pass or fail, the logistic regression predicts a probability of the establishments fail this food inspection. It is up to user to identify where to create the cutoff value between 0-1 to predict a pass and predict a failure. In this case, we considered a 40% cutoff, and *Figure 4* compare the two threshold on the distributions of both the training and testing results. Our accuracy for the 40% cutoffs was 84%, and all the detailed results are presented in *Table 3.*

Figure 5 displayed how the model's predictions compare to the actual inspection results from 2016 testing data. Figure 6 showed that instead of simply think inspection predictions as pass or fail, we could also interpret "failed" from 0 to 1 based on our needs.For three neighborhoods to be tested separately, Hyedepark got highest AUC value (0.96) while Chinatown got lowest AUC value (0.886), which means our model works better in Hyedepark neighborhood.

Discussion

Above all, our model only contains 10 productive predictors, which make the data needed to operate this project more manageable. But there are limitations in this Chicago Smart Food model, the model is build based on the food inspection results from last year, then it's more likely to be useful and accurate in predicting the near future. If we need to predict in a long time run, we may need to build model based on time series theory. Another problem here is that when we run cross validation on our model, the accuracy standard deviation is higher than we expected, which means our model is overfitting. And we will adjust our model to fix this problem and probably show the result in final video.

Conclusion

Predictive analysis is more frequently to be used in sociological and urban planning topics in recent decades. The predictive results helped governments to be smarter, cheaper and better at serving residents, and our Chicago Smart Food application is designed to make these data and predictive results to be more accessible to citizens. More importantly, our Chicago Smart Food application could inspectors much more convenient finish their inspections and supervisors more efficiently manage this food inspection system

Figure 1 Project Process

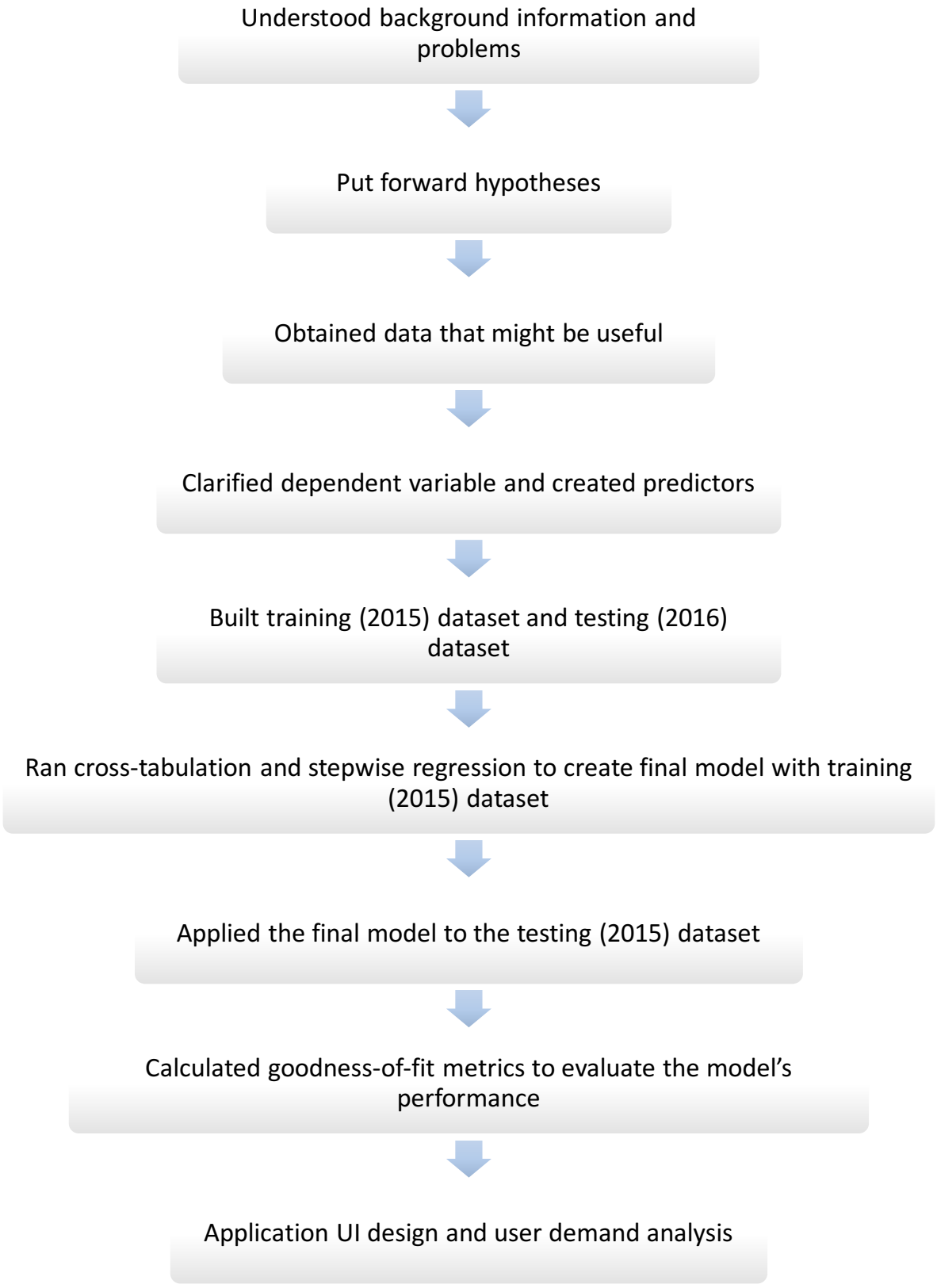


Table 2 Predictors in the final regression model

	Variable	Estimate	Std. Error	z value	Pr(>|z|)
1	(Intercept)	-5.8830	0.1468	-40.0688	0.0000
2	1IS_LAST_FAIL	0.2287	0.0928	2.4634	0.0138
3	EXPIRE	-0.3114	0.0980	-3.1775	0.0015
4	VIOLATION	-0.2363	0.0854	-2.7684	0.0056
5	COUNT	2.1432	0.0462	46.4073	0.0000
6	JAN	1.3755	0.1586	8.6718	0.0000
7	FEB	1.7330	0.1485	11.6727	0.0000
8	MARCH	1.4846	0.1414	10.4969	0.0000
9	APRIL	1.3744	0.1429	9.6196	0.0000
10	MAY	1.6319	0.1351	12.0805	0.0000
11	JUNE	1.1361	0.1388	8.1826	0.0000
12	JULY	1.1735	0.1412	8.3132	0.0000
13	AUGUST	1.1004	0.1368	8.0411	0.0000
14	SEP	0.8033	0.1359	5.9093	0.0000
15	OCTOBER	0.9388	0.1295	7.2463	0.0000
16	NOVERMBER	0.5906	0.1306	4.5237	0.0000
17	BURGLARY	0.0434	0.0232	1.8681	0.0618
18	TRESPASS	-0.0199	0.0116	-1.7214	0.0852
19	ARSON	0.2394	0.1588	1.5072	0.1318
20	HOMICIDE	0.3241	0.1640	1.9758	0.0482
21	WOODLAWN	1.5843	0.3285	4.8226	0.0000
22	HYEDEPARK	1.0111	0.2226	4.5417	0.0000
23	GARBAGE	0.0478	0.0133	3.5994	0.0003

Figure 2 Received Operating Characteristic (ROC) Curves

Figure 2(a). ROC for Training (2015) Data

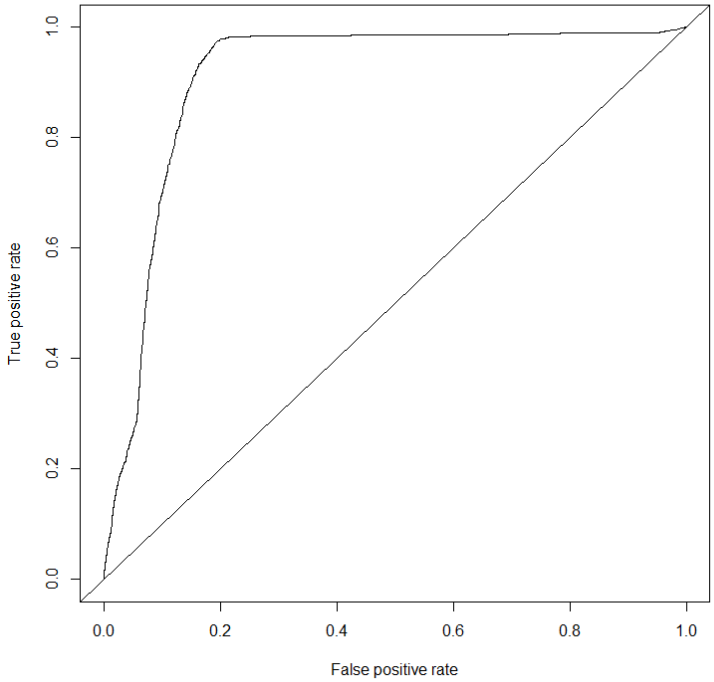


Figure 2(b). ROC for Testing (2016) Data

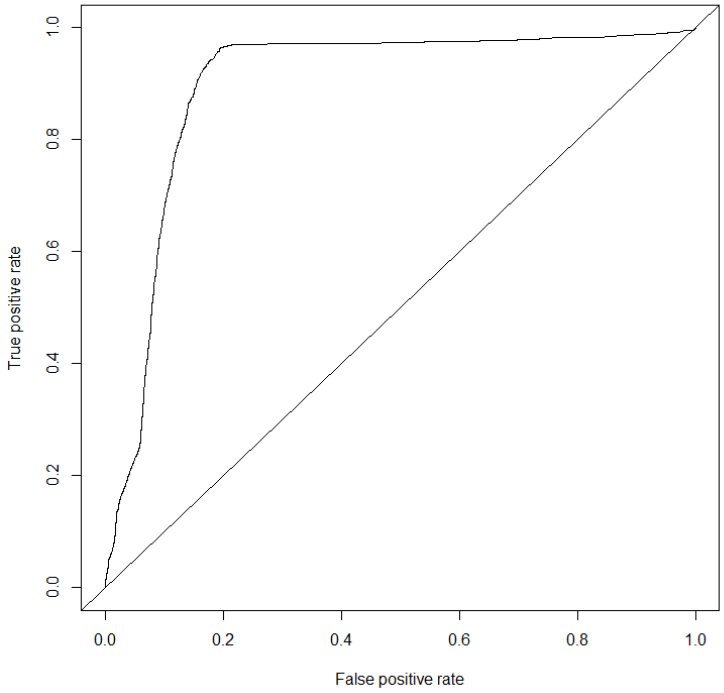


Figure 3 Probability Distribution Plots

Figure 3(a). Probability Distribution Plot for Training (2015) Data

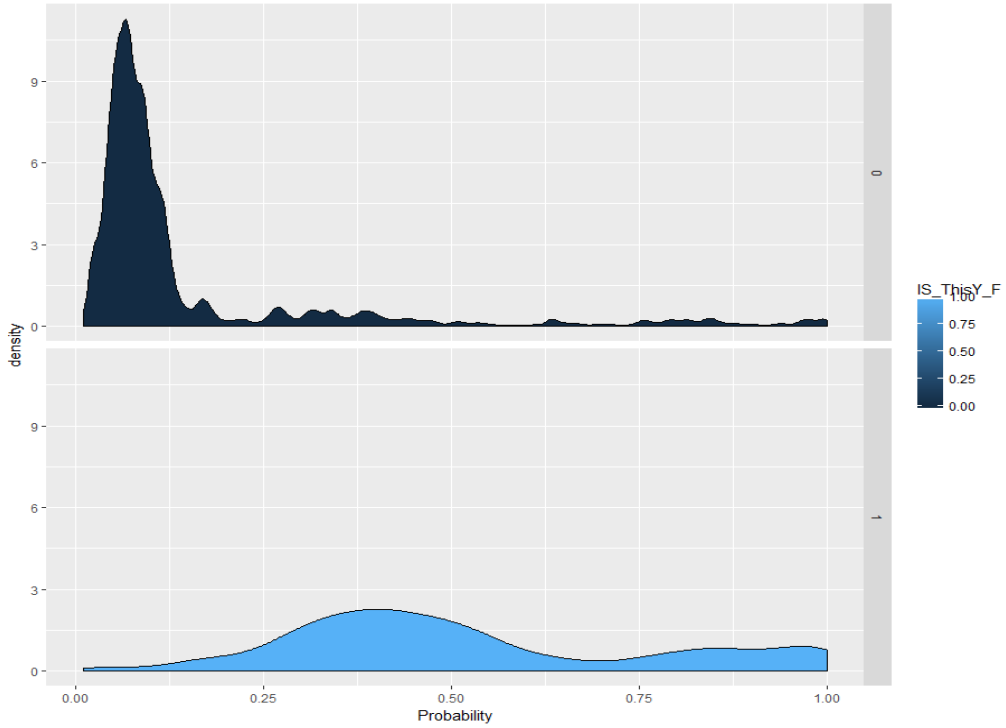


Figure 3(b). Probability Distribution Plot for Testing (2016) Data

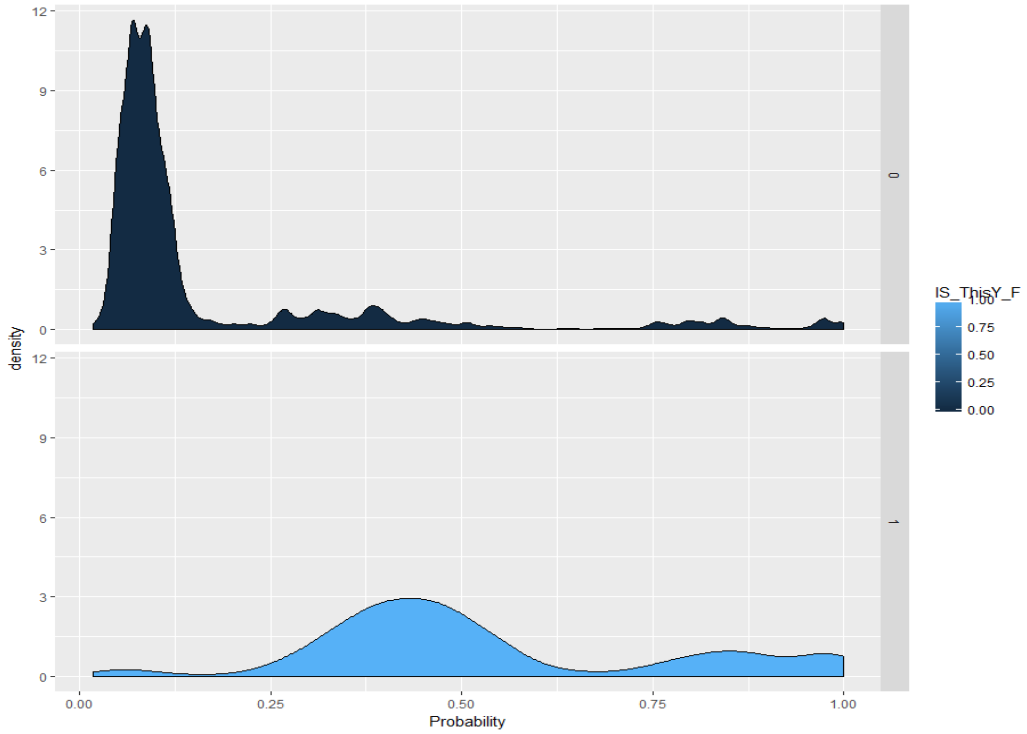


Table 3 Goodness-of-fit Indicators

	Training Data Set	Test Data Set
McFadden's Pseudo-R^2	0.323	0.318
Area Under the ROC Curve (AUC)	0.9098	0.8974

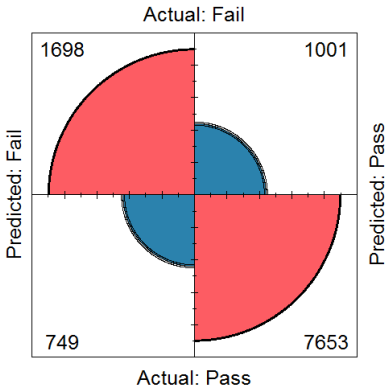
Confusion Matrix

Failed = probability of 0.4 or greater

	Actual Pass	Actual Fail
Predict Pass	7653	1001
Predict Fail	749	1698
Hit Rate of 84.24%		

	Actual Pass	Actual Fail
Predict Pass	6434	845
Predict Fail	731	1858
Hit Rate of 84.03%		

Fourfold Pplots
for 40% Threshold Confusion Matrices



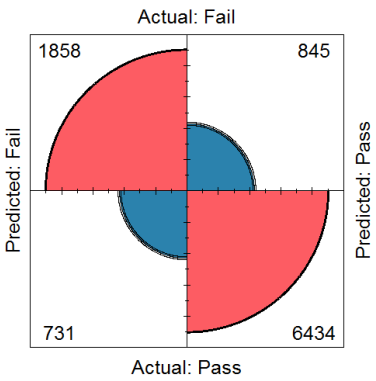


Figure 4 Actual and Predicted Inspection in 2016

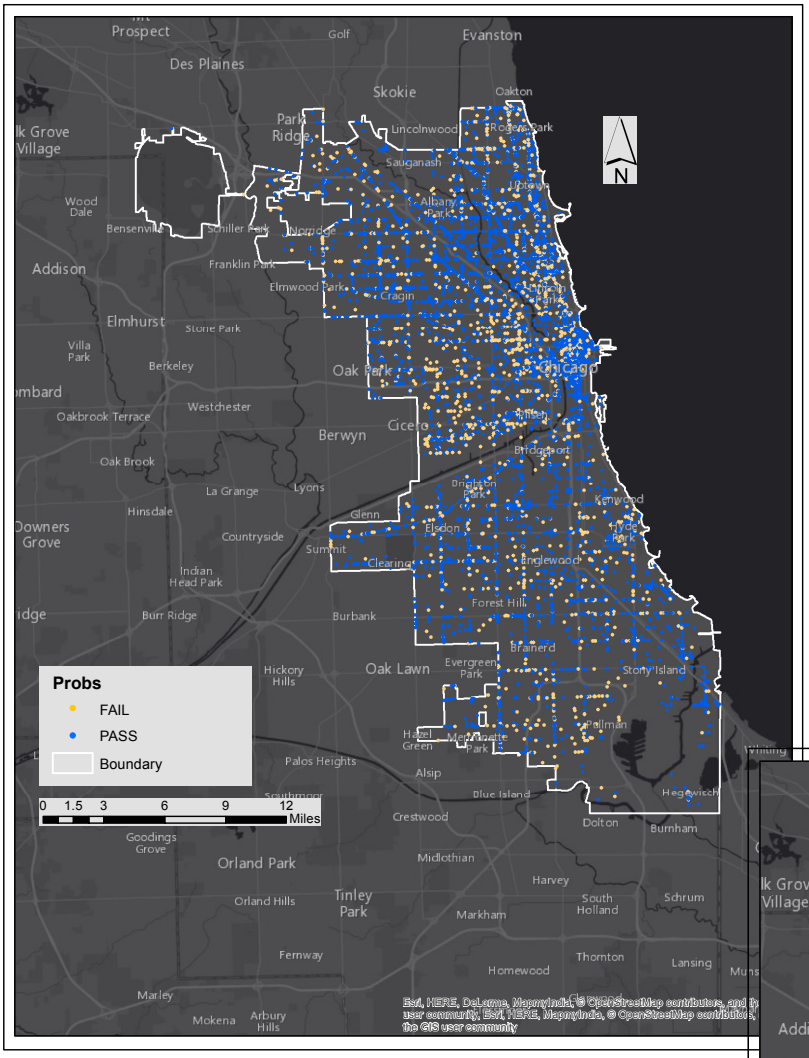


Figure 4(a). Actual Inspection Result in 2016

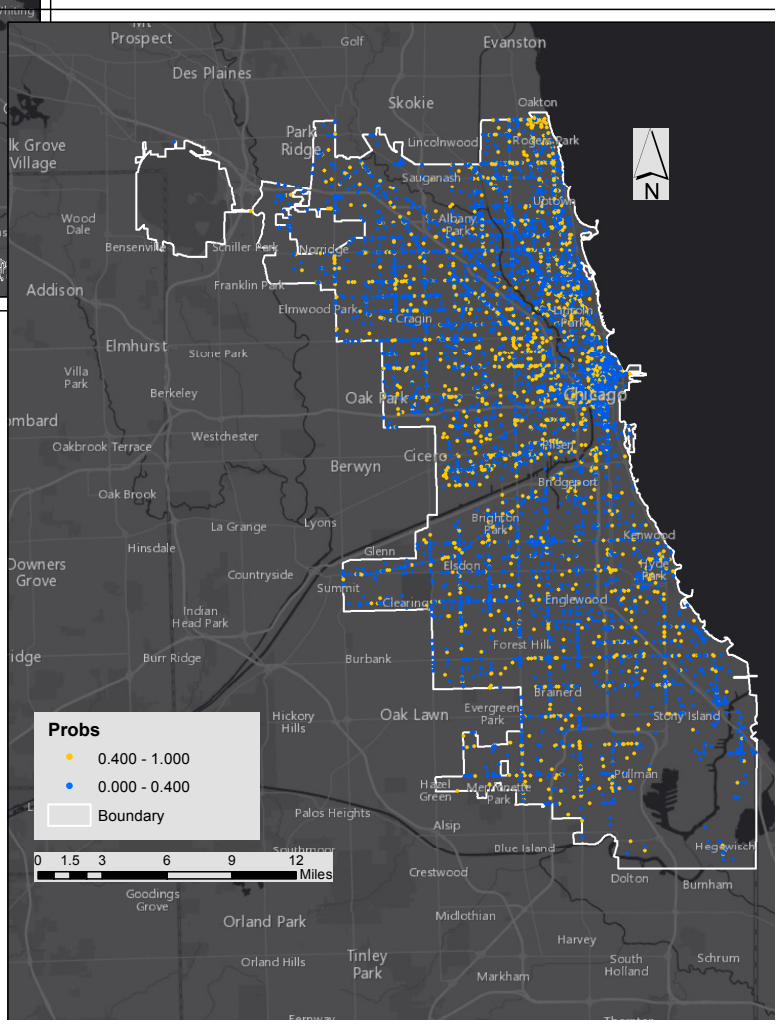


Figure 4(b). Predicted Inspection Result in 2016 (with 40% failure threshold)

Figure 5 Predicted Inspection Outcomes in 2016

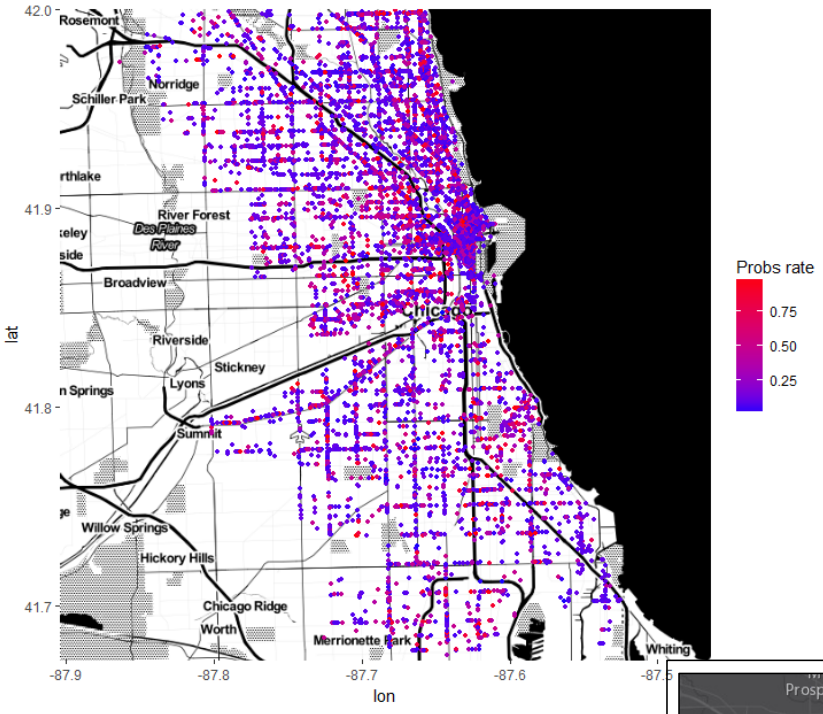


Figure 5(a). Predicted Inspection Outcomes Made by R in 2016

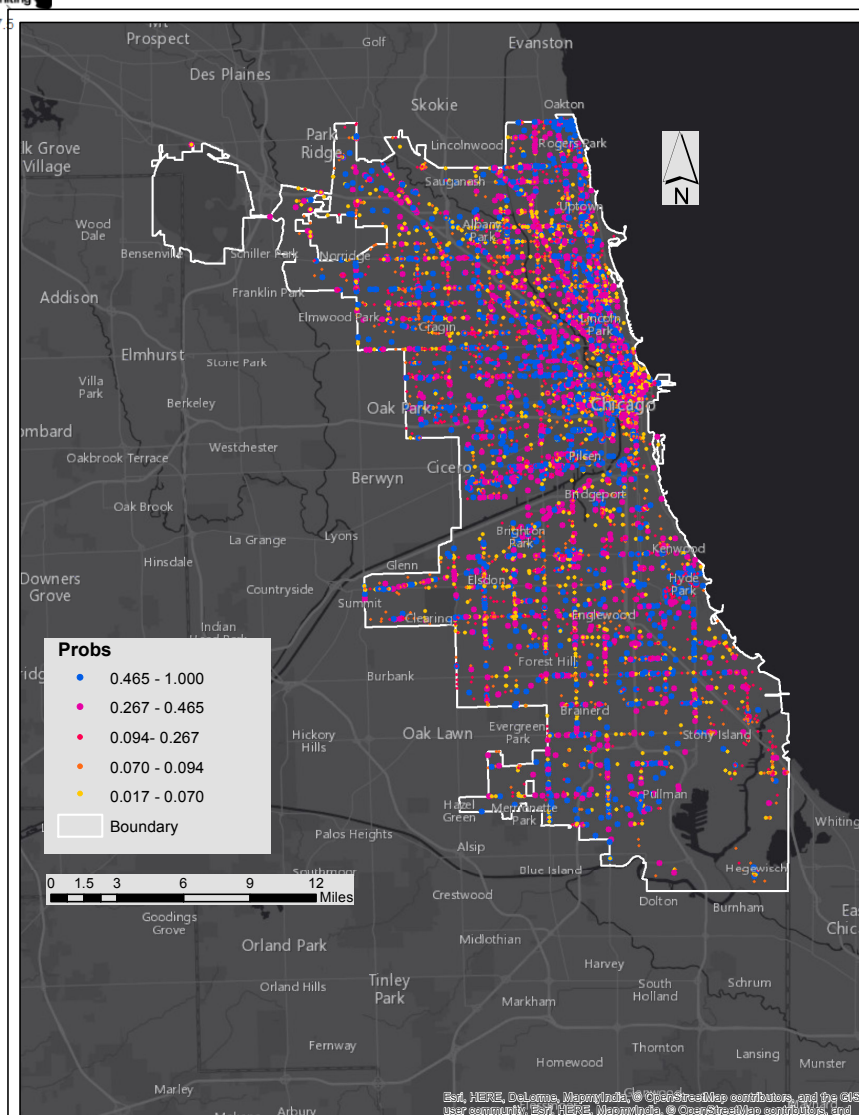


Figure 5(b). Predicted Inspection Outcomes Made by ArcGIS in 2016