University of St. Gallen

Group project DataScience Fundamentals

The prediction of hygiene ratings for food stores in New York City

Matthias Steiner, André Ruckdäschel, Kaspar Lichtsteiner

supervised by
Johannes Binswanger and Juan-Pablo Ortega

02 December 2019

Contents

1 Introduction

It is estimated that every year 48 million people fall sick, 128'000 are hospitalized and 3'000 die from food born illnesses (for Disease Control & Prevention, 2018). These numbers clearly illustrate the necessity of clear and strict oversight over any organisation handling large quantities of food, thusly constantly operating under the danger of poisoning a significant part of the population. Among the four governmental agencies working on minimizing food posed risks, the 'Food Safety and Inspection Service' handles (among other things) the inspections of food retailers (Food Safety and Inspection Service, 2018). The agency employs over 9'000 people and presides over an annual budget ranging from 1,03 - 1,05 billion USD, with the cost of its field operations in the State of New York alone totalling 13,3 - 15,3 million USD each year (U.S. Census Bureau, 2018, p. 16)

Based on this information, we conclude that successful predictions of which food retailers pose the highest public health risk and the subsequently more efficient allocation of resources, could not only lead to cost reductions of millions of USD every year but might even save lives by preventing the outbreak of dangerous foodborne illnesses. This paper details the data gathering, as well as the application of different analytical methods, aimed to predict the inspection grade of food retailers in NYC.

2 Date Cleaning and Preparation

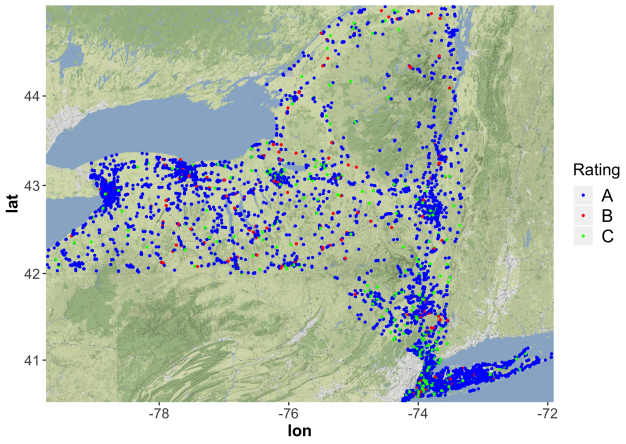
2.1 New York Inspection Data

The main dataset for our project is retrieved from the official site of the State of New York. It contains a total of 28'300 A to C ratings of food store inspections together with basic information of the stores as well as deficiency descriptions.

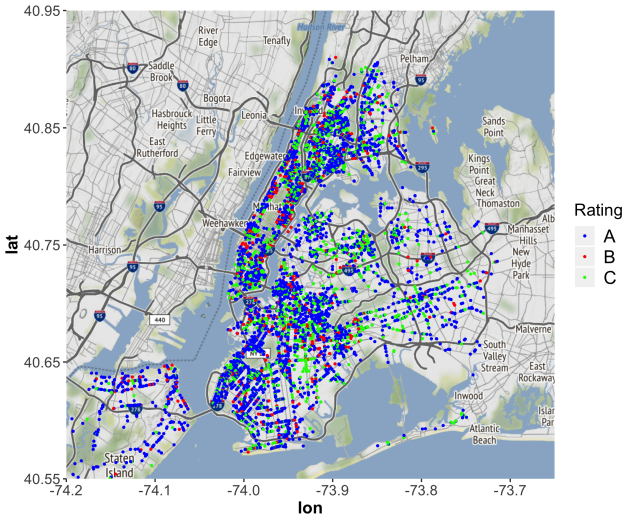
Our analysis predicts the food store inspection ratings from this dataset only for the City of New York (counties New York, Kings, Bronx, Richmond and Queens). We decided to focus merely on city level due to the vast availability of covariate data. In addition, the distribution of classes is highly imbalanced on state level which can be diminished by focusing on city data only (more about this issue in section 3.1).

Figure 1: Geographic Distribution of the inspected Shops

(a) New York State



(b) New York City



2.2 Chain Information

We used information about the owner from the data to identify if the shop is a chain as well as the number of other stores from our data that are part of the same chain. Presumably, chains have better processes in place to achieve a high hygiene standards.

2.3 Spatial Data

Latitude and longitude information is embedded in a larger address string. We create a function to extract the location data in two new columns. A check for `NA` values reveals that 748 observations have no location information. The exact address, however, is available for all shops. Therefore, we provide the address to the Google Maps API[1] to retrieve the missing latitude and longitude details except for 198 observations that were dropped. A Map of the completed data is illustrated in Figure 1 on state and city level.

With complete spatial information, we now compute the two variables "shops density in 1km radius" as well as "rating of the closest neighbor". The distances of coordinates in meters are

[1]In essence, the Google Maps API is a free service offered by Google. It requires only a one-time registration with a valid email address. Afterwards, it generates an API Key that must be included in the R-script with the command `register_google(key = "API KEY")`.

calculated with the Haversine Formula[2]

$$d = R\left(2\,\text{atan2}(\sqrt{a}, \sqrt{1-a})\right) \qquad a = \sin^2(\frac{\Delta\alpha}{2}) + \cos(\alpha_1)\cos(\alpha_2)\sin^2(\frac{\Delta\lambda}{2})$$

where α_i are latitudes, λ_i are longitudes and R is the earth's radius ($6371 \times 10^3 m$).

2.4 Subway Data

We amend the New York City inspection data with location information of subway stations. We use again an official dataset provided by the State of New York. The dataset contains the location of every subway station. We apply again the Haversine Formula to find the distance of the closest station to every shop.

2.5 Demographic Data

Since the original dataset did not feature any demographic data, related to the inspected stores, this type of data had to be acquired otherwise. Fortunately, the amount of demographic data on different sections of the state New York is ample, in the form of different datasets detailing the results of various census-endeavours. We judged the data provided by the U.S. census bureau to be the most reliable, mainly due to the large sample sizes, that their estimates were based on. Based on their data, two datasets, one conveying demographic information by U.S.-county and one by Census Tract, were created (US Census Bureau, 2017).

Whilst the first dataset could be directly merged via the county-variable, direct merging with the second dataset was not possible, since Census Tracts do not follow the traditional pattern of location specification used in the inspection?s dataset. An approach of nevertheless matching the sets, was found in the creation of a third ?translation? dataset AddTrac (short for Address to Census Tract) which matched the retailers? locations with the Census Tract codes. To do so, we made use of the census bureau?s geocoding service (US Census Bureau), Geocoder).

[2]The Haversine Formula has its roots in spherical trigonometry and calculates the geodesic distance – the shortest path between two points on a curved surface of a sphere, like the earth. It needs to be mentioned that Haversine Formula does not take into account changes in altitude (Van Brummelen, 2013, pp. 157 - 160). However, it provides sufficient accuracy for the scope of this project.

This service added various geolocation-identifiers to specifically created csv files holding the addresses of our food retailer. Unfortunately, circa 20% of the addresses were not identified by the service (possible reasons include name changes and confidentiality concerns (U.S. Census Bureau, 2018, p. 7)) which led to some data loss. The address specific Census Tract Ids were ultimately created by combining the output of state FIPS , county FIPS and census block codes, as well as adding placeholder zeros where necessary.

2.6 Google Web Scraper

While doing research to find datasets with regards to food inspections we decided to obtain more information from webpages. The reviews from customers, written on google places, could give us information about how individual persons perceived their visit to the specific food shop. In order to gather this information from the internet we set up a web scraper. We used the package *RSelenium* to execute an automated google search for all the observations. This included a lot of trial and error due to unforeseen changes of the xPaths and other errors. Most of the errors only showed up in the middle of the scraping process, which had a duration of about 24 hours for all the food shops. Therefore, it was only possible to adjust the function after the script run during the night. This resulted in a lot of waiting and adjusting but finally the data was scraped successfully.

A brief analysis of the dataset revealed the incompleteness of the ratings. Approximately 40% of the shops don't have an entry in google places and thus no rating. After a lot of discussion on how to handle the problem we agreed on using the number of reviews as a parameter of the internet popularity and assign all the missing shop a value of 0.

2.7 Airbnb Data

In order to acquire additional data on the location of the shops, we integrated data from Airbnb (average price and number of rooms). To join the original and the airbnb data frames, a matching key is needed. We use latitude and longitude to assign every Airbnb observation to a ZIP code. Thereafter, we grouped the data by ZIP code, calculated the respective means and counts and added it to the new tibble.

3 Data Analysis

3.1 The Class Imbalance Problem

Closer scrutiny of Figure 1 sheds light on the extremely imbalanced distribution of the original data. In fact, the rating A contributes to 64% on city level while B and C are only represented by 9% and 27% respectively. This differences in the occurrence has a large impact on the prediction capabilities of a model. The univariate linear discriminant analysis (LDA) illustrates the issue. The LDA decision rule and boundaries are

$$\delta_k(x) = x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + log(\pi_k) \Longrightarrow \delta_1(x) = \delta_2(x) \Leftrightarrow x = \frac{\mu_1 + \mu_2}{2} + log\left(\frac{\pi_2}{\pi_1}\right)\frac{\sigma^2}{\mu_1 - \mu_2}.$$

If the two classes are perfectly equally represented, the prior probabilities are $\pi_1 = \pi_2$ and the log term gets zero. In the presence of highly skewed classes, in turn, π_1 and π_2 are different from each other. If class 1 occurs more often it holds that $\lim_{x\to 0^+} log(x) = -\infty \Leftrightarrow \lim_{\pi_1 \to 1 (\pi_2 \to 0^+)} log(\frac{\pi_2}{\pi_1}) = -\infty$. So, the following three extreme cases for decision boundaries can be distinguished:

$$x = \lim_{\pi_1 \to 1} \frac{\mu_1 + \mu_2}{2} - \infty \qquad x = \frac{\mu_1 + \mu_2}{2} \text{ with } \pi_1 = \pi_2 \qquad x = \lim_{\pi_1 \to 0^+} \frac{\mu_1 + \mu_2}{2} - \infty$$

Hence, an imbalance of the two classes moves the decision boundary infinitely to the right or to the left respectively. As a consequence, the model will always predict the class that is heavily overrepresented and ignore the others. The methodology presented in the next chapter will exactly address this issue and improve the prediction accuracy of inspection grades B and C.

3.2 Analysis Methodology

James et al. (2017, p. 316) used bagging to reduce the variance in tree models. However, instead of taking a random subset of the entire data like in normal bagging, we used repeated subsets that show an equal distribution of the three classes. Therefore, we either needed to take more observations from the B or C classes (oversampling) or remove a part of the observations from the A class (undersampling). Fernández et al. (2018, p. 83) outlines that both approaches have drawbacks. While oversampling increases the likelihood for overfitting since we create exact copies of a minority class, undersampling ignores potentially useful data points. To address these weaknesses, we combined over- and undersampling with bagging which results in over- and under-bagging (OU-bagging). In detail, we applied the following procedure:

1. Take a subset of the data where the three classes are balanced either with over- or undersampling

2. Use K-Fold Cross-Validation (CV) to get cross-validated errors from this subset

3. Repeat the entire process B times and then take the average rate of the CV-errors to get the bagged-CV-errors

After we found the model with the lowest bagged-CV-error, we estimated the model B times and took a majority vote of all the B models to classify the observation to one of the three categories.

3.3 Pre-Analysis Variable Selection

The obvious drawback of the described methodology is that it is computationally intensive. Using all 100 variables would be accordingly difficult since we know that the tuning of a boosting model with 100 variables would take many hours. Furthermore, some of the demographic variables are linear combinations of each other and must be excluded from our analysis anyway. For the remaining ones, we calculated the correlation to the Inspection Grade we wanted to predict and took the 20 covariates with the highest correlation.

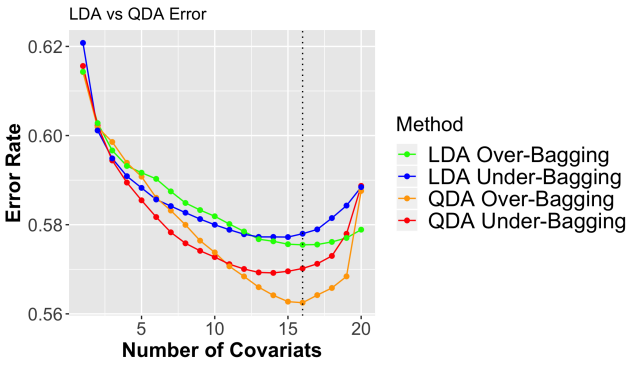
3.4 Linear and Quadratic Discriminant Analysis

Our first method is the discriminant analysis. We combined the discussed OU-bagging technique with a variable selection approach as suggested by James et al. (2017, pp. 205 - 210). First, we created a function for *Best Subset Selection* which estimates a model for all possible variable combinations. Together with OU-bagging as well as CV we would have estimated a total of approximately 1 billion models per learning technique ($B \times K \times 2^p$ with $p = 20$, $K = 10$ and $B = 100$). The computation capacity of our machines as well as the given time constraints made this approach not feasible for us. Instead, we decided to use *Forward Stepwise Selection* which requires an estimation of 190'000 models in total ($B \times K \times \frac{p(p+1)}{2}$).

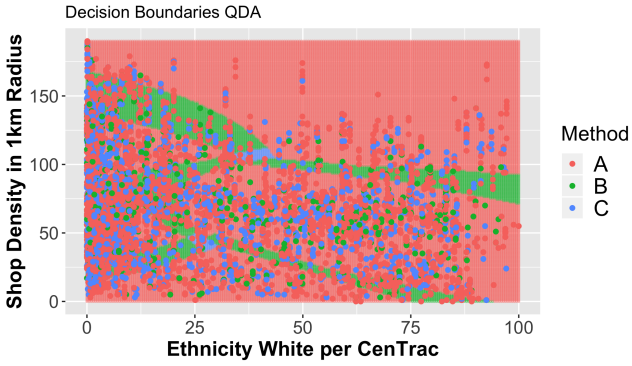
Figure 2 presents the results from this analysis. Panel (a) shows the bagged-CV-error of all estimated models with 1 to 20 covariates. QDA with over-sampling and 16 covariates performs best. However, the prediction error still amounts to over 50%. When we implemented the model

Figure 2: Geographic Distribution of the inspected Shops

(a) LDA vs QDA Errors



(b) Decision Boundaries QDA



suggested by Panel (a) and used it for our data, we achieved an error of 54%. Panel (b) illustrates the QDA decision boundaries in a bivariate setting with "shop density" and "Ethnicity White per Census Track". A relatively large amount of observations is assigned to category B which partly explains the poor prediction accuracy. The prediction matrix from Table refpred-matrix supports that assumption since a large part of A are classified as B. This reveals one of the already mentioned drawbacks of our approach because over-sampling copies B observations multiple times and therefore overweights their proportion.

Table 1: Prediction Matrices of Different Estimated Models

QDA			
Obs / Pred	A	B	C
A	2839	462	1164
B	1260	173	594
C	228	26	103

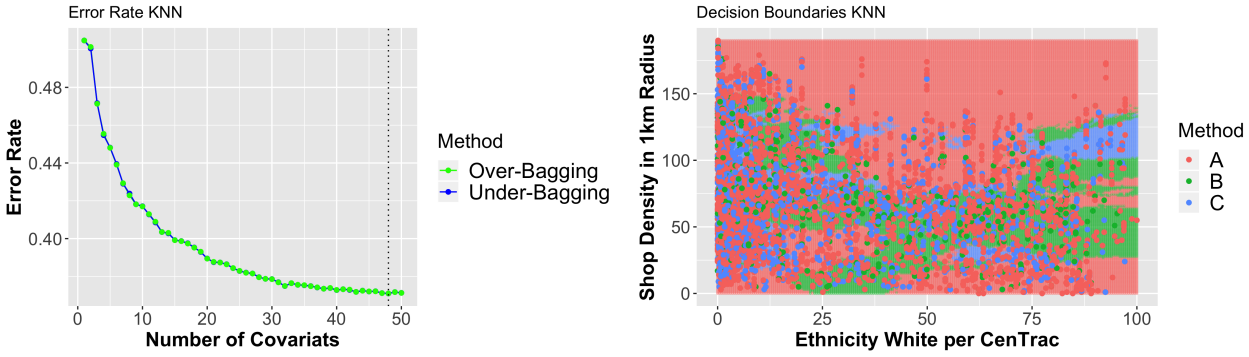
KNN			
Obs / Pred	A	B	C
A	1464	204	470
B	1928	335	885
C	935	122	506

Random Forest			
Obs / Pred	A	B	C
A	4327	0	0
B	0	661	0
C	2	0	1859

3.5 K-Nearest Neighbor

Next, we implemented K-Nearest Neighbor (KNN). First, we tried to implement KNN similar to the discriminant analysis and adjusted our backward stepwise selection function for that purpose.

Figure 3: Geographic Distribution of the inspected Shops



(a) KNN Errors

(b) Decision Boundaries KNN

However, a restriction in the KNN function only allows up to 1000 neighbors which creates an error for discrete variables with low variance. Therefore, we used all covariates instead and estimated the error for different k again with OU-bagging.

Panel (a) in Figure 3 shows the estimated bagged-CV-error for different k. The lowest error is achieved with a k of 48. OU-bagging performed de facto equally well. We estimated the best model with k equal to 48 and over-bagging. The resulted error rate of 66% is quite higher than the estimated bagged-CV-error of 37%. The prediction matrix of Table 3.4 shows that knn again overweights the minority classes B and C.

3.6 Random Forest Analysis

We decided to use the random forest analysis to have a model with a typically good performance but relatively little tuning requirements. Additionally, the structure of the dataset according to the correlation matrix, seemed rather complex and not linear. Tree models normally outperform linear models in these situations James et al. (2017)[S. 314] In order to address the problem of class imbalances, the random forest model is built up on the bagging theory mentioned above. Panel (c) shows the results of the majority vote prediction for the oversampling. As mentioned above, it is very likely overfitting and therefore probably will not perform with this accuracy in

new data sets. Although there is normally only little tuning required, the evaluation of tuning parameter is computationally very intensive due to an exponential rise of possible combinations with every new tuning parameter. Therefore, we were restricted to a smaller selection of possible tuning grid and only used 40 different combinations.

Figure 4: A variable importance plot for the selected parameters. Variable importance is computed using the mean decrease in Gini index and expressed relative to maximum

(a) Under sampling

(b) Over sampling

The importance of the variables can be shown in figure 4. The distance to the next subway station is in both samples the most important variable. Therefore, it is also used in the Individual Conditional Expectation analysis.

Figure 5: Individual Conditional Expectation (ICE) for over- and under sampling

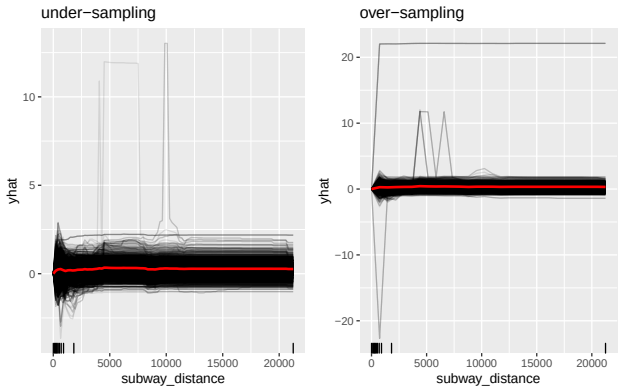


Figure 5 shows that even the most important variables do not have a significant impact on the predictions in the random forest. model. Thus, additional and more relevant parameters are needed in order to achieve a better performance.

3.7 Gradient Boosting

As a further method of tree based models we implemented extreme gradient boosting. It enables a better performance than random forest and provides more advanced tuning methods. The selected parameters have 576 different combinations which have to be evaluated. The current approach with a full hyper grid search is computationally very intensive and takes several days with an ordinary computer. Another possibility is to use a random discrete search which tests only a sample of it with a pre-defined stopping parameter instead of all combinations. Thus, with the given time constraints this approach was not feasible for us.

4 Conclusion

Despite the utilization of an array of different variables and methods, the prediction of inspection grades proved to be exceedingly difficult. We believe that the causes for a certain rating can be found in the characteristics of the restaurants themselves and not certain attributes of the neighborhood like we used. Therefore, parameters like the age of the owner, his education or what kind of products are offered could have been more valuable predictors. In addition, the imbalance in the classes further complicated the estimation of accurately predicting models. The over- and under-bagging was computationally too intensive and did not lead to the expected results. In a similar situation we would, therefore, try to overcome the imbalance issue with other means like the more advanced Synthetic Minority Oversampling Technique (SMOTE) that artificially creates new observations of the minority data.

References

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from imbalanced data sets*. Cham, Switzerland: Springer. doi: https://doi.org/10.1007/978-3-319-98074-4

for Disease Control, C., & Prevention. (2018, November). Estimates of foodborne illness in the united states. *Burden of Foodborne Illness*. Retrieved from `https://www.cdc.gov/foodborneburden/2011-foodborne-estimates.html`

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An introduction to statistical learning. with applications in r* (8th ed.). New York Heidelberg Dordrecht London: Springer.

Van Brummelen, G. (2013). *Heavenly mathematics. the forgotten art of spherical trigonometry*. Princeton and Oxfordshire: Princeton University Press.