Predicting Yelp Ratings of Toronto Food Establishments

Melissa Cañas
Nivi Lakshminarayanan
Jaime Ramirez-Cuellar
Gavin Tieng
PSTAT 135: Big Data Analytics
Dr. Adam Tashman
December 11, 2020

Abstract

Our project answers the question, “Is a restaurant or establishment’s rating on Yelp influenced or predicted by its government health inspections and other attributes?” We decided to narrow our project scope to solely analyzing data from Toronto, since health inspection data is easily accessible. Our response variable was establishment rating, which was categorized into “bad”, “neutral”, and “good”. Our predictor variables were severity of infraction, status of inspection, actions taken, the number of reviews the business has received from Yelp users, whether or not the restaurant has a take-out option, whether or not the restaurant has a delivery option, whether or not the restaurant has outdoor seating, whether or not the restaurant has bike parking, and the price range of the restaurant. To model our data, we used logistic regression, random forest, and the Naive Bayes classifier. After model evaluation, we chose the random forest classifier to be the best model. We conclude that there were too many null observations in our dataset to make entirely accurate predictions and future steps would have to be taken to determine which factors accurately predict an establishment’s Yelp rating.

Data Overview

Our project utilizes data from both Yelp and the Toronto government DineSafe database. The public Yelp database holds a wide variety of datasets, and we decided to use the “yelp_academic_dataset_business.json” Yelp dataset, renaming it as “business.json.” This file has a very organized schema outlining all of the pieces of information contained. The dataset holds general location data such as the business name, address, city, state, zip code, and even latitude and longitude coordinates, as well as other information such as business attributes (e.g. open for take out, type of parking accessible, etc…), genre of cuisine, business hours, and the number of reviews the business has received from Yelp users. Most importantly for us, the dataset contains information regarding the establishment's star rating, rounded to the nearest half-star. This was the primary piece of information used in our data analysis of the Yelp dataset.

DineSafe is Toronto Public Health's food safety inspection program that monitors all establishments serving and preparing food. Every establishment receives a minimum of one, two, or three inspections each year depending on the specific type of establishment, the food preparation processes, volume, and type of food served. Each establishment in Toronto must prominently post the most recent inspection notice near the establishment's main entrance.

We obtained DineSafe's data from Toronto's open data server in XML format; we used Python's libraries xmltodict and json to transform the data into JSON format. The dataset was slightly less organized than the Yelp data and required more pre-processing to become suitable for our purposes. After doing so, the final DineSafe dataset included general location data such as food establishment name, address, longitude, and longitude, as well as inspection information such as status of the inspection ("pass", "conditional pass", or "closed"), severity of infraction ("minor", "significant", "crucial", and "not applicable"), actions taken ("Notice to comply" and "Corrected during inspection"), and more. This information would be used in future modeling.

Methods

Data Pre-Processing

The first step was to clean the health inspection data from DineSafe in the notebook "dineSafeEDA_preprocessing.ipynb". We first filtered the data by eliminating entries without an address or business name, as well as making every address and business name lowercase to ease with matching once we joined the Yelp data with DineSafe. We also converted the "longitude" and "latitude" parameters that were originally string types to float types. We then created separate dataframes organized by their combination of whether or not an inspection had occurred and if any infractions were received by the food establishment to take a closer look at the information relevant to each case. We then joined these separate dataframes to create the final DineSafe dataframe, making sure that all schemas were the same, and saved the data in a parquet file "dineSafeParsed.parquet".

It was then time to create our response variable. To do so, we created a 'rating' column from the stars variable from the Yelp data in the notebook "YelpEDA_preprocessing.ipynb", choosing to group businesses with 2.5 stars or below as "bad", businesses between 2.5 and 4 stars as "neutral", and businesses with 4 or more stars as "good." We also further filtered the Yelp data to just analyze restaurants from Toronto, and accomplished this by filtering using the "city" parameter included in our Yelp schema. We further filtered the Yelp data by eliminating data entries with no address or no business name. We also made every address and business name fully lowercase, again to ease with matching.

Finally, we joined the two datasets using different methods in the "YelpEDA_preprocessing.ipynb" notebook, ultimately deciding that joining by name and address was the most proficient. We initially considered joining by the "longitude" and "latitude" parameters of the Yelp schema and matching the restaurant coordinates with those in the health inspection dataset, but noticed that the Yelp longitude and latitude data was written to 9-10 decimal points, and was at a much higher level of granularity than the health inspection longitude and latitude data, instead at 4-5 decimal points. There were also slight differences in the decimal points between both datasets, so truncating the Yelp longitude and latitude points to match the health inspection points failed to be a valid option. Joining at this level yielded as few as 6 matches, so we decided instead to join by both name and address, with our final dataset having

1,401 observations. We saved this new dataset as a parquet file “data.parquet” to preserve this new schema.

Exploratory Data Analysis

The vast majority of establishments passed their inspections. Out of 17,644 establishments, approximately 78% of them passed all of their inspections in the analysis period; whereas 13.4% of the establishments conditionally passed at least one of their inspections. Closed establishments are rare – only 33 establishments closed. This may indicate that whether or not an establishment passed their inspection would not be a very meaningful predictor when trying to fit our model. Most of the inspected establishments are restaurants (41%), food take outs (15.7%), and food stores (convenience/variety; 13.3%) – see Figure 1.

Figure 1: Number of Establishments by Type

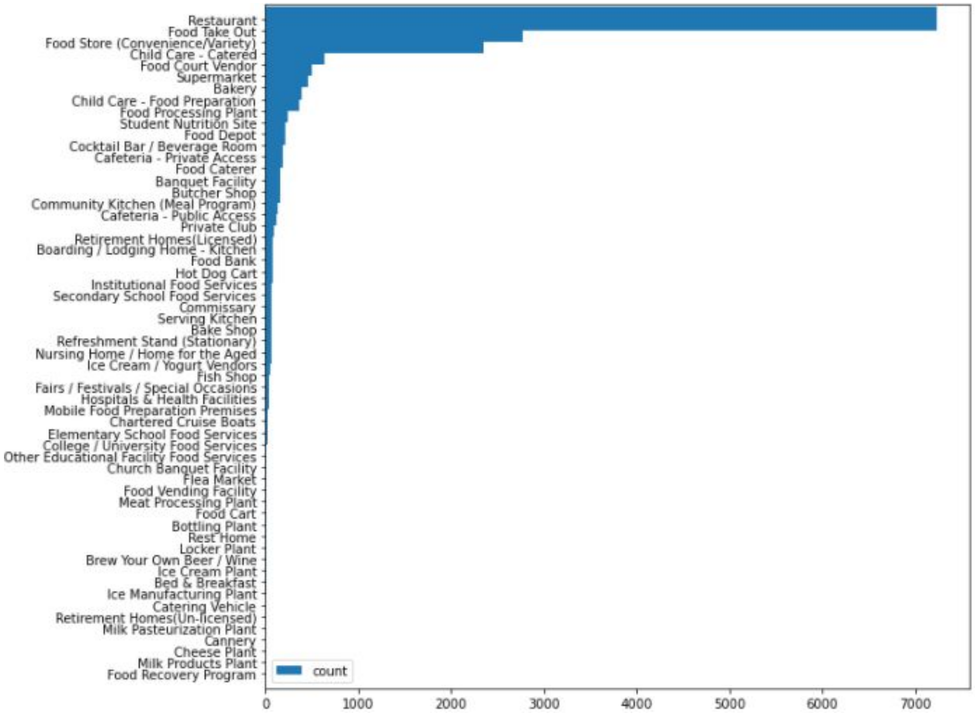
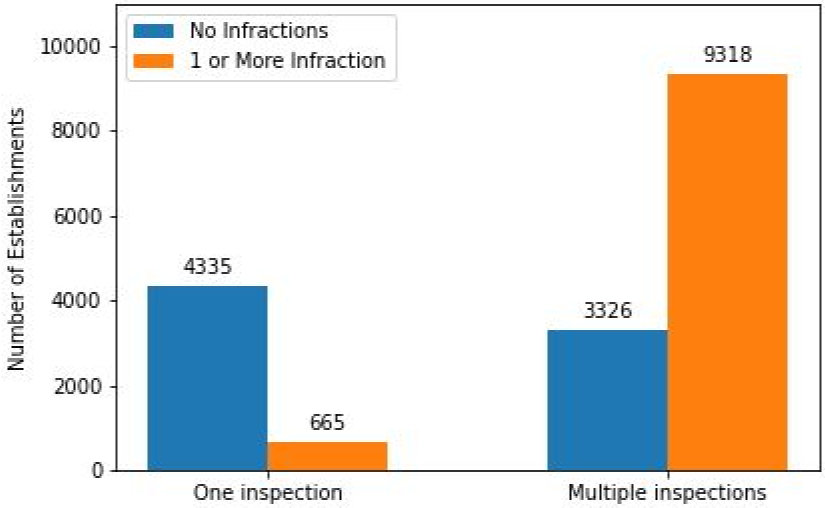
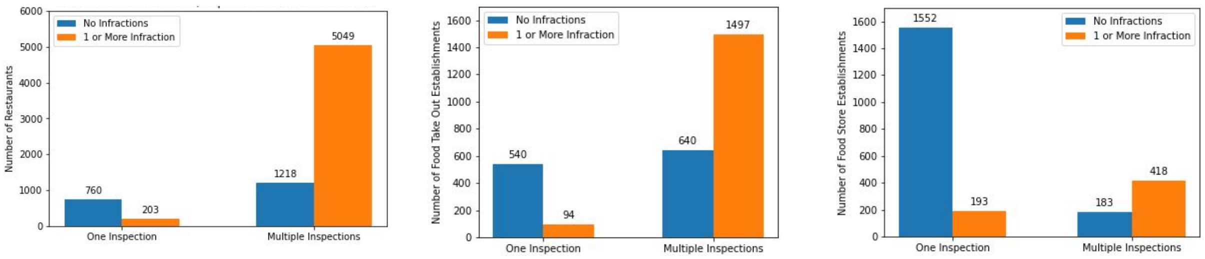


Figure 2: Distribution of Infraction and Inspections Across Establishments



As we can see from Figure 2, for the establishments that receive only one inspection, about 87% of them had no infractions, while the remaining 13% of them had an infraction. Additionally, for the establishments that received more than one inspection, approximately 74% of them had 1 or more infractions, while the remaining 26% had none.

Figure 3: Distribution of Infraction and Inspections Across Establishment Type



a. Restaurants

b. Take out establishments

c. Food store

Figure 3 shows the number of establishments that receive or not infractions by number of inspections and type of establishment. For instance, in Panel a, the distribution of restaurants that fall into the 4 categories closely follows the distribution across all establishments. This may tell us that restaurants will be a strong predictor of other information. Additionally, this makes sense considering "restaurant" makes up roughly 41% of all establishments recorded. Similarly, we see that the distribution of infractions/inspections on food take out establishments (Panel b) follow a similar distribution to the distribution across all establishments. Food take out establishments account for approximately 16% of all recorded establishments. Finally, looking at the distribution of infractions/inspections across food store establishments (Panel c), we can see that roughly 74% of them did not receive an infraction. However, out of the remaining 25% who received at least one inspection, approximately 70% of them received 1 or more infractions. This is interesting, because it doesn't quite follow what we saw in the distribution across all establishments, and might indicate that food store establishments aren't a good predictor. Furthermore, food store establishments make up 13% of all establishments.

Preparing Data for Modeling

We performed all of our modeling in the notebook "ModelBuilding.ipynb". After further exploration, we learned that the data had a large number of null observations. As a result, we decided to go about dealing with the missing values in two different ways to see which one would lead to more accurate predictions. The first option was to drop all observations with null values. The second option was to replace all null values with the corresponding column's mode.

To choose our predictor variables, we went through a process of trial and error to see which ones would be most relevant to our question and which ones had at least 50% non-null values. Our final predictor variables were severity of infraction, status of inspection, actions taken, the number of reviews the business has received from Yelp users, whether or not the restaurant has a take-out option, whether or not the restaurant has a delivery option, whether or not the restaurant has outdoor seating, whether or not the restaurant has bike parking, and the price range of the restaurant. We would have liked to include more predictors, but including variables with more than 50% of observations being null values ended in lower accuracy in our models. We then transformed the data so that our predictor variable values in each observation would be a single feature vector.

Lastly, we randomly split the dataset into training and testing data where 80% of the dataset was used to train the selected models and 20% of the dataset was used to test model performance. As a result, when dropping all null values, our training data consisted of 289 observations and the testing data consisted of 76 observations. When replacing all null values with the column's mode, the training data consisted of 1106 observations and the testing data consisted of 295 observations.

Model Construction

When running the models, we found that dropping null values results in better predictions, so those procedures are what we will be discussing. However, running the models when replacing the null values with the corresponding column's mode had very similar procedures.

We decided on running logistic regression, random forest, and Naive Bayes classification models on our data. We decided on these three specific classification models as we needed models that would be able to accurately work with multinomial classification, since our model would be predicting a rating of 0, 1, or 2, showing that our model is not meant to be binary. A linear regression model did not make sense to use since our dependent variable was not continuous. We decided on the random forest model since we knew that having multiple single trees based on a random sample of the training data would be more accurate than using a single decision tree model. However, since random forest models can result in overfitting due to their large size, we decided to also use a model with a lower size, and that is the Naive Bayes.

We first modeled our data using logistic regression. To do so, we used the LogisticRegressionWithLBFGS function from the mllib.classification library to fit the model on our training dataset. We then made our predictions using the testing dataset.

Then, we used the random forest classifier to model the data using RandomForestClassifier from the ml.classification library, using 9 trees to fit the model on our training dataset. We again then made our predictions using the same testing dataset.

Lastly, we modeled the data using the Naive Bayes classifier. We did so using the NaiveBayes function from the ml.classification library, with a smoothing parameter of 0.5 and a multinomial model type to fit the model on our training dataset. We then made our predictions using the same testing dataset.

Results

The following chart describes the accuracy rate resulting from the three models when dropping null values and imputing null values:

Model	**Accuracy - Dropped Null Values**	**Accuracy - Imputed Null Values**
Logistic Regression	0.618421	0.447458
Random Forest	0.671053	0.555932
Naive Bayes Classifier	0.552632	0.427119

The accuracy rates of the models where the nulls are imputed with the respective column’s mode are consistently lower than those from the models where the nulls are dropped. Because the accuracy of the random forest classifier when dropping null values is the highest, we choose that to be the best model for our data.

Conclusion

To answer the question, “Which factors are good predictors of Yelp ratings?”, we chose to analyze the Yelp business dataset and the Toronto DineSafe dataset, narrowing our search to the Toronto area. We modeled the data using logistic regression, random forests, and Naive Bayes classifier models, including several establishment attributes and inspection information as our predictors. While we chose the random forest classifier to be the best model as a result of having the lowest accuracy rate, we believe that our dataset having a large amount of null values and removing them as a result, ended in predictions that were not as accurate as they could have been.

In order to improve our model in the future, it might be appropriate to narrow our search to a different area besides Toronto, where the data is not so sparse, so we can include more establishment attributes as predictors. Additionally, we could include the “yelp_academic_dataset_reviews.json” Yelp dataset, which includes information about all reviews that businesses have received from Yelp users. By doing so, we can use information about reviews, such as the average review length, number of positive and negative words used, and percentage of uppercase letters in the model as well.

References

https://open.toronto.ca/dataset/dinesafe/

https://www.yelp.com/dataset

https://www.yelp.com/dataset/documentation/main