Predicting Italian Restaurant Count in Chicago Neighborhoods through Classification

Maxwell Burner

I. Introduction: A Simple Question

For my project, I chose to start one of the prompts given in the assignment description: "In a city of your choice, if someone is looking to open a restaurant, where would you recommend they open it?" In particular, I decided to investigate which neighborhoods in Chicago are most suitable for opening an Italian Restaurant, and how can a neighborhoods suitability for an Italian restaurant be predicted based on the presence of other venues?

Business Question

Restaurants and bars are ubiquitous, with most reasonably large cities having a variety of different dining venues to choose from. All kinds of different restaurants exist based on different cuisines (Chinese, Japanese, Mexican, Ethiopian, etc. for many more), or on different dietary preferences or categories of food (grill restaurants, seafood restaurants, vegetarian restaurants). While conventional wisdom often states that up to 90% of restaurants fail within the first year of opening, actual research (Forbes Magazine) has shown that in reality new restaurants show failure rates of only 17%.

Obviously a restaurant's success depends heavily on factors such as layout, quality of food, and quality of service. Still, restaurants often co-occur, clustering in certain parts of a given city. What factors determine the kinds of locations favored by restaurants? What kinds of traits are found in neighborhoods with larger or smaller numbers of restaurants?

I chose to focus my research on the city of Chicago for several reasons. First, Chicago is a large, well known city, and thus an attractive place for opening a new restaurant. Second, my methodology results in analyzing a dataset with entries for each neighborhood of the city, so the large number of neighborhoods in Chicago allows for a large dataset; a larger dataset in turn translates to more robust models and statistical analyses. As far as type of restaurant I chose to keep my aims simple, at look at one of the most ubiquitous varieties; the Italian restaurant.

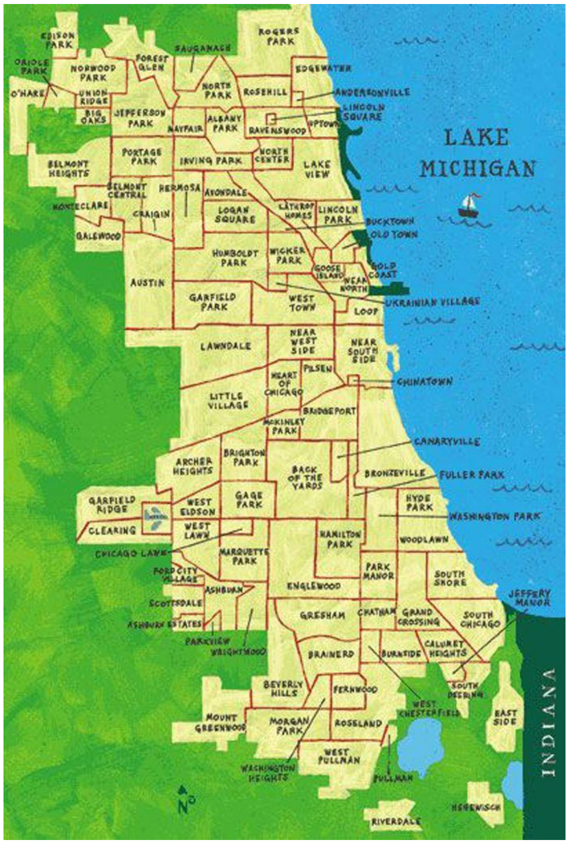
II. Data and Usage

My data comes from four sources.

Wikipedia

First, I used the Wikipedia page for Chicago neighborhoods () to obtain lists of neighborhoods in Chicago and the city Community Area (simply called 'districts' in the notebook) to which they belong.

The data provided by Wikipedia effectively has two attributes: city community area (‘district’) and city neighborhood. The nature of the webpage (some entries in the table were links, others were not) made web scraping impractical, so I manually created a list of districts and lists of neighborhood by district. With these I constructed a small database with an entry for each neighborhood, with columns for district name and neighborhood name.



Geocoding

My second source is coordinate data accessed through the Nominatim object provided by the geopy/geocoders library for Python. This provides a means of obtaining coordinates corresponding to each neighborhood identified by Findwell. The data comprises simply of coordinates, paired values corresponding to Latitude and Longitude.

Foursquare

My final source is FourSquare, which I can use to obtain information on venues near each set of coordinates. Foursquare data from a venues request has an entry for each venue returned, with various attributes and information on each venue. The relevant attributes that will be extracted include venue name, name of venue category, and venue location coordinates. A neighborhood and district will also be assigned to each venue.

Each venue in the dataframe of Foursquared data can be assigned (with reasonable accuracy) to whichever neighborhood corresponds to the request that returned that venue. This will allow for the creation of a dataframe of venue counts by neighborhood, with a row for each neighborhood and attributes corresponding to each venue category.

III. Methodology

A. Database Creation

1. Neighborhood Location Dataframe

I started by manually creating a python dictionary in which each key was the name of a district, and each value a list of the names of the neighborhoods within the key district. Using Geocoding and a function I wrote for this purpose, I used each entry in the former dictionary to create a list of dictionaries, with entries for district name, neighborhood name, neighborhood latitude, and neighborhood longitude. Tuples of latitude and longitude were added to a set, and any neighborhood whose coordinates were not already in the set (to eliminate repeats) got a dictionary in the list. These lists were each appended to an initially empty dataframe with columns for district name, neighborhood name, neighborhood latitude, and neighborhood longitude; the end result was that the dataframe had an entry for each neighborhood in Chicago.

2. Venue Dataframe

The dataframe of neighborhood locations was fed into another function of my own creation, to obtain venue data from Foursquare. The function made a call for each neighborhood in the input dataframe, requesting all venues within 5km of the neighborhood coordinates. Data from the returned JSON file was extracted to create a dataframe with a row for each venue returned by the call, and attributes for name, category, neighborhood, district, latitude, and longitude. The function also stored the ID of each returned venue in a set, and checked the venue ID of each new venue to exclude any already present in the set; this prevented the creation of multiple entries for a single venue.

3. Venue Count Dataframe

A vector was created containing each unique value obtained for venue category. For each neighborhood in the neighborhood dataframe, for each venue category in the aforementioned vector, the number of venues of that category present in that neighborhood were counted up and recorded. This led to the creation of a dataframe with a row for each neighborhood and an attribute for each venue category, with the intersection of the *i*th row and *j*th column being the number of venues of category *j* in neighborhood *i*.

B. Predicting Italian Restaurant Count with Classification

1. Predictor Value and Target Values

The predictor variable matrix was generated as a matrix containing all the values from the venue count data frame, excluding the column corresponding to Italian Restaurant counts.

The target variable was the number of Italian restaurants in each neighborhood. This was observed to have values of 0, 1, 2, 3, 4, 5, or 6. Because it had discrete values, the Italian restaurant count was encoded into a categorical variable using two approaches. First, a series was made with entries for each neighborhood and values mapped from Italian Restaurant Count as follow: zero Italian restaurants became 'Very Poor', 1 Italian restaurant became 'Poor', two became 'Okay', three became 'Good', four became 'Very Good', five became 'Excellent', and six became 'Perfect'; this was the Italian Restaurant Rating Vector (IRRV).

Another series was also created with entries for each neighborhood. Again, values were derived from Italian restaurant counts, but this time using a 'Boolean' encoding strategy; Italian restaurant counts of zero or one were mapped to a value of 0, while counts from two to six inclusive were mapped to a value of 1. This is referred to from here one as the Italian Restaurant Boolean Vector (IRBV).

2. Selection of Machine Learning/Modeling Methods

Again, in either the single-label or multi-label case the target variable was categorical/discrete. It therefore made little sense to try and apply regression models; only classification models were used. Initial trial runs with various forms of classification model, and consideration of the benefits and downsides of each algorithm, led to three classification methods being tested.

3. K-Nearest Neighbors Classification

The *K*-nearest neighbors classification method was applied because it is one of the simplest and most universally applicable classification modeling techniques. Unlike the other

a. Mutli-Label

Validation of K-Value

For every value of *K* from 1 to 19, a K-Nearest Neighbors classification object was created using the neighbors module from SciKit-Learn. For each of these classifiers, the steps of (1) splitting data (with IRRV as target variable) into training and testing data (with 30% of data used for testing), (2) fitting the classifier to the training data, (3) using the classifier to predict based on testing predictor data, and (4) calculating accuracy score (subset accuracy, calculated using a function from SciKit-Learn) from predicted test values and observed test values, were repeated 200 times. A dataframe was created showing K-value, and corresponding mean, median, standard deviation, and skew of the 200 calculated accuracy values. The dataframe was examined, and a *K*-value chosen as the best option based mainly on mean and standard deviation of the accuracy score.

Testing of Model

Another *K*-nearest neighbors classifier was created, using the chosen value of *K*. This model was fitted to the entire dataset (rather than undergoing train-test splitting), used to predict target variable (IRRV), and the accuracy score was calculated using the vector of predicted values and observed IRRV.

b. *Single-Label*

The validation and testing stages were repeated, this time using the Single-Label IRBV as the target variable.

4. Decision Tree Classification

Decision tree classification was applied mainly because this approach provided the option of visualizing the resulting tree, making it possible to see the 'decision-making process; the visual makes it possible to understand which predictors were used by the classifier, and how they were used to predict target variable.

Multiple Labels

The approach used above to validate different value of *K* for the *K*-Nearest Neighbors classifier was used again to validate different maximum tree depths for a Decision Tree Classifier, testing maximum depths from 1 to 19.

A tree of the chosen maximum depth was tested for model accuracy as was done for the K-Nearest Neighbor Classifier. This final tree was visualized, showing the nodes and connections, to reveal what factors were used to make decisions about predicted value.

Single-Label Approach

The steps taken above for a multi-label decision tree were repeated with a decision tree predicting a single label target, but with the same changes made for validating K-value for single-label *K*-nearest neighbors.

5. Logistic Regression Classification

Logistic Regression Classification was applied because when used for single-label predictions it allowed for use of Log Loss as an additional metric to quantify accuracy.

Multi-Label Approach

The approach used above to validate *K*-values for *K*-nearest neighbors classifier was used with some modifications to validate different values of *C* (the inverse of regularization strength); the values tested were 1, 10, 15, 20, 30, 50, 100, 1000, and 2000. The Logistic Regression classifiers were fit to the IRRV, using the 'newton-cg' solving method and the 'ovr' method for dealing with multiple class labels.

Single-Label Approach

The approach used for validating *C* for multi-label logistic regression was repeated for single label logistic regression; that is, the target variable was IRBV rather than IRRV. In this case, log loss was also calculated for each model fitted, and the final database reported mean and standard deviation for accuracy score and log loss.

C. Comparing Neighborhoods by Target Variable Values

1. Counting

For both the multi-label rating of neighborhoods, and the one-label (Boolean) rating of neighborhoods, the number of neighborhoods with each rating value were counted.

2. Correlations

First, each venue category other than Italian restaurant was assessed to determine how number of venues of that category correlated with number of Italian restaurants in a neighborhood; this was accomplished by generating a correlation matrix for the venue counts dataframe, then extacting the column corresponding to Italian Restaurant count. The ten venue categories (other than Italian Restaurant) that most strongly correlated with Italian restaurants were noted.

3. Grouping and Assessing Top Venues

Neighborhoods were grouped by Italian Restaurant Rating (Very Poor, Poor, Okay, Good, Very Good, Excellent, or Perfect). In each group, the mean number of venues for each category were calculated, and the five most common venues in that group were recorded. This was repeated, but grouping neighborhoods by Italian Restaurant Bool Vector values (0 or 1) instead of Italian Restaurant Rating Vector.

Neighborhoods were again grouped by IRBV values, and the mean prevalence of each venue category calculated for each both groups. For each venue category, the difference in mean prevalence between group 1 and group 0 was calculated and reported, to determine the venues whose prevalence differed most between the two groups.

D. Mapping Neighborhoods

Using the Folium Python library, two maps were created of the Chicago area, both with markers for each neighborhood in the database. One map colored markers by the IR Rating value assigned to that neighborhood: red for ‘Very Poor’, orange for ‘Poor’, yellow for ‘Okay’, green for ‘Good’, blue for ‘Excellent’, and purple for ‘Perfect’. The second map colored neighborhood labels according to each neighborhoods value in the IR Bool vector: red for 0, green for 1.

IV. Results

A. K-Nearest Neighbors Classification

1. Multi-Label

Validation of K-*Value*

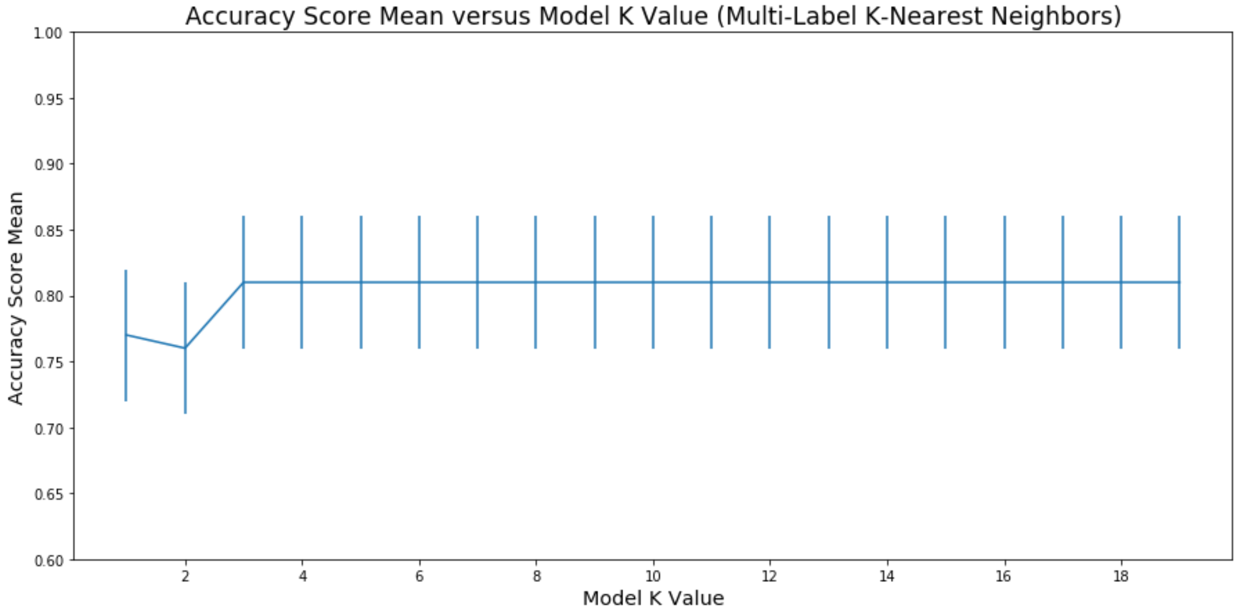


Figure 1: Accuracy Score Mean versus Model K*-value for a Multi-Label* K*-Nearest Neighbors Classifier.* Error bars represent standard deviation of the accuracy score. Means and standard deviations were calculated from 200 trials with random assignment of training and testing data.

The mean accuracy score obtained by a *K*-Nearest Neighbors classifier was 0.77 for *K* = 1, 0.76 for *K* = 2, and 0.81 for *K* > 2. Standard deviation was consistent at 0.05 across *K* values.

Testing Accuracy

When a *K*-nearest neighbors classifier with *K* = 3 was fit to the entire data set, then used to predict IR Ratings for the whole data set, comparison of predicted *y*-hat and observed *y* produced an accuracy score of 0.85.

2. Single Label

Validation of K-*Value*

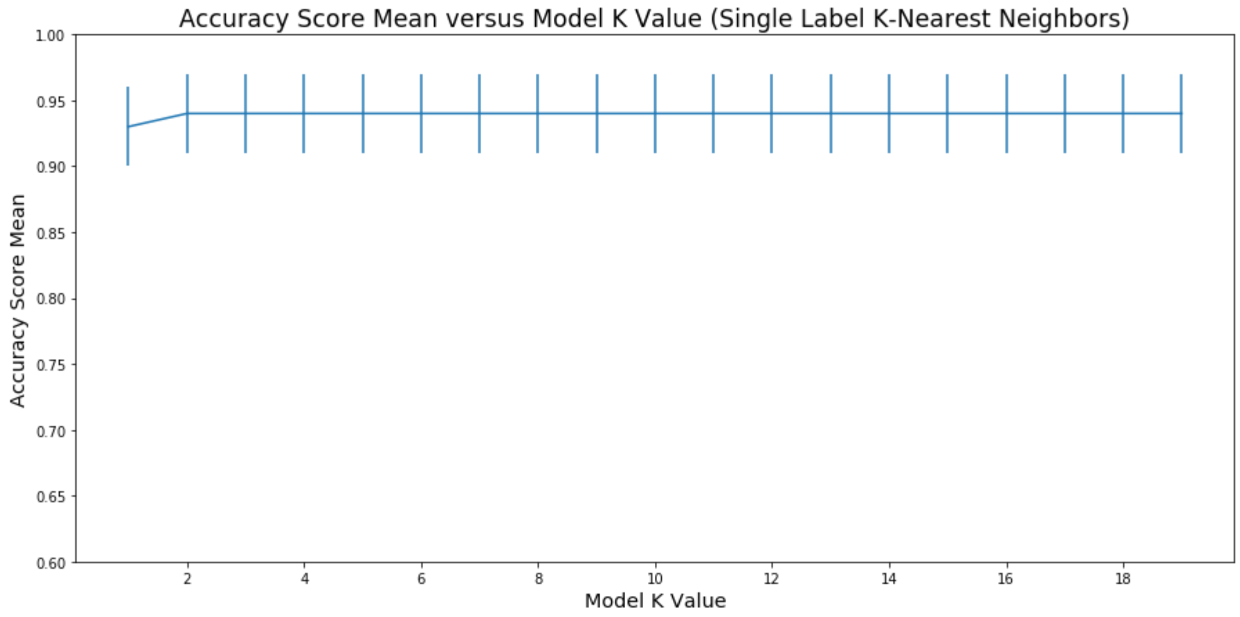


Figure 2: Accuracy Score Mean versus Model K*-value for a Single-Label* K*-Nearest Neighbors Classifier.* Error bars represent standard deviation of the accuracy score. Means and standard deviations were calculated from 200 trials with random assignment of training and testing data.

Observed mean accuracy scores were 0.93 for a *K* value of one, and 0.94 for all *K* > 1. Standard deviation was 0.05 across all values of *K*.

Testing Accuracy

When a *K*-nearest neighbors classifier with *K* = 3 was fit to all predictor data and the single-label IRBV values, and told to predict single-label ratings based on all predictor data, a comparison of the predicted and observed values yielded an accuracy score of about 0.94.

B. Decision Tree Classification

1. Multi-Label

Validation of Tree Depth

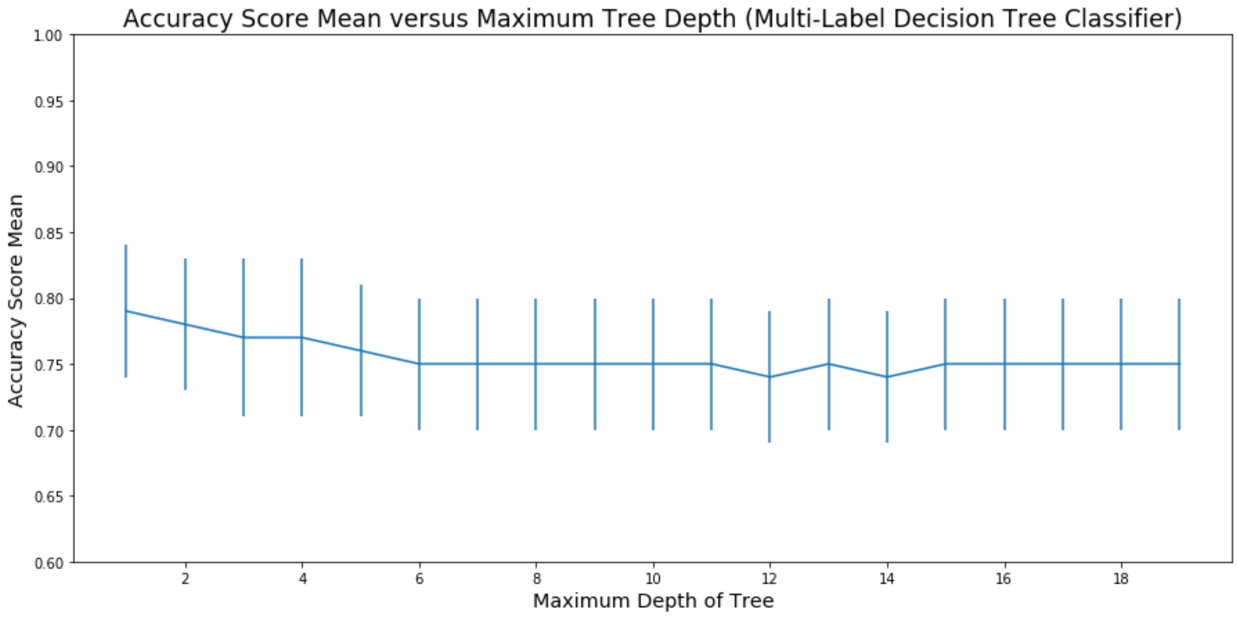


Figure 3: Accuracy Score Mean versus Maximum Tree Depth for a Tree Classifier fit to Multi-Label Data. Error bars represent standard deviation of Accuracy Scores. Means and standard deviations were calculated over 200 trials with random assignment of training and testing data.

Mean accuracy score for a decision tree fit to multi-label data was highest at 0.79 for a depth of one; it subsequently decreased with increasing tree depth until reaching at 0.75 for maximum depth of 5; for greater depths the mean oscillated between 0.75 and 0.74. Standard deviation was consistent at 0.05.

Confirmation of Accuracy

When a decision tree with a maximum depth of 2 was fit to all of the multi-label data, and told to calculate a predicted *y*-hat from all of the predictor data, comparing the predicted *y*-hat and observed *y* gave an accuracy score of 0.816.

Examination of Tree Structure

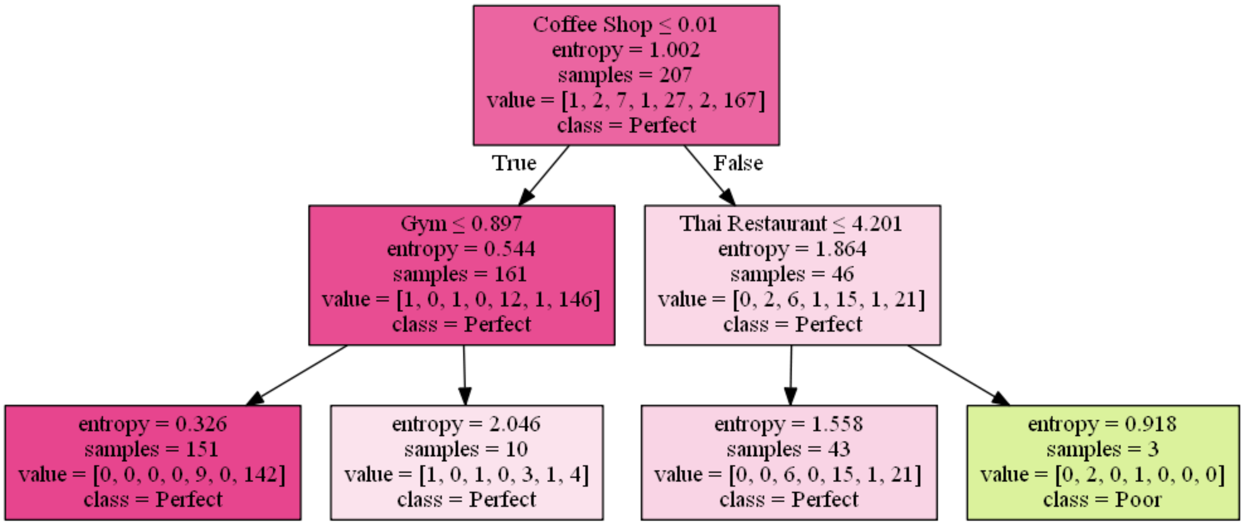


Figure 4: Tree Structure for Decision Tree (Maximum Depth of 2) Predicting Multi-Label Italian Restaurant Rating of Neighborhoods.

As shown the tree split data first by prevalence of coffee shops in a neighborhood, then by prevalence of gyms in one branch and prevalence of Thai restaurants in the other branch. Neighborhoods falling into the leaf with Coffee Shop ≤ 0.01 and Thai Restaurant ≤ 4.201 were predicted to have a rating of 'Poor' (one Italian restaurant), while neighborhoods falling in all other leaves were predicted to have a rating of 'Perfect' (six Italian restaurants).

2. Single-Label

Validation of Tree Depth

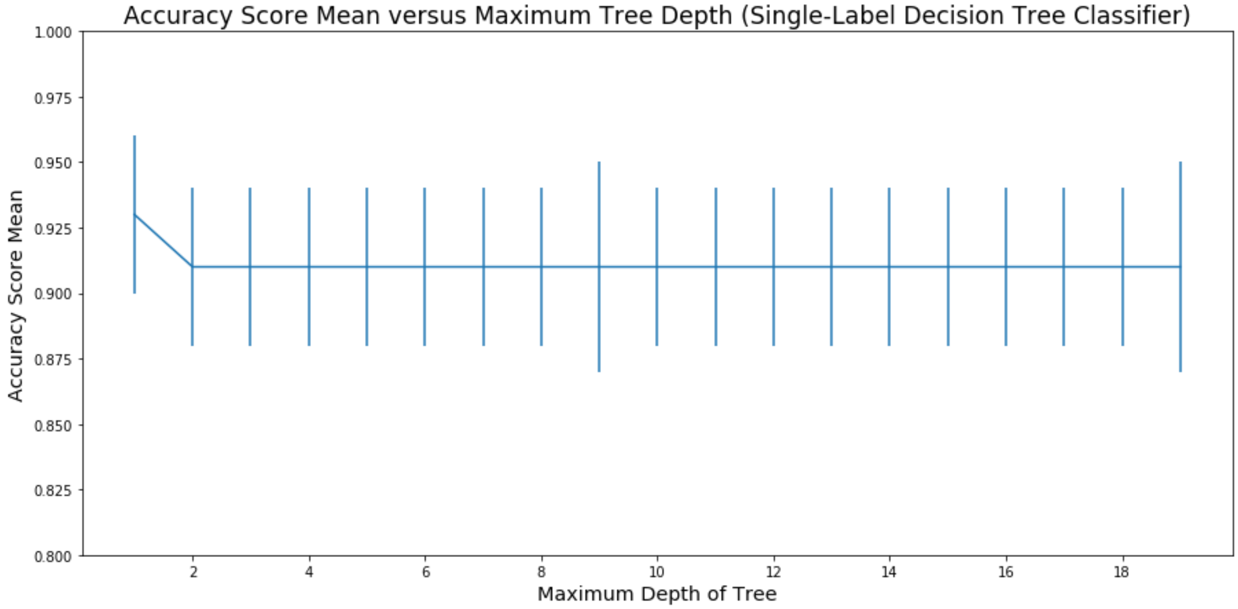


Figure 5: Accuracy Score Mean versus Maximum Tree Depth for a Decision Tree Classifier fit to Single-Label Data. Error bars represent standard deviation of Accuracy Scores. Means and standard deviations were calculated over 200 trials with random assignment of training and testing data.

For a decision tree fit to single-label data the mean accuracy score was highest at 0.94 for a maximum tree depth of one; for all maximum depths above one the mean accuracy score observed was 0.90, or 0.92 when maximum depth was 10. Standard deviation of the accuracy score was mostly consistent at 0.03 across all tested values, with some maximum depths producing a value of 0.04.

Confirmation of Accuracy

Although mean accuracy in tests was highest at a depth of one, it was judged that a depth of two was preferable so that the number of leaves would be closer to the number of labels.

When a decision tree classifier with a depth of 2 was fit to all of single-label data, and used to calculate predicted *y*-hat from all of the predictor data, comparing observed *y* and predicted *y*-hat resulted in an accuracy score of 0.96.

Examination of Tree Structure

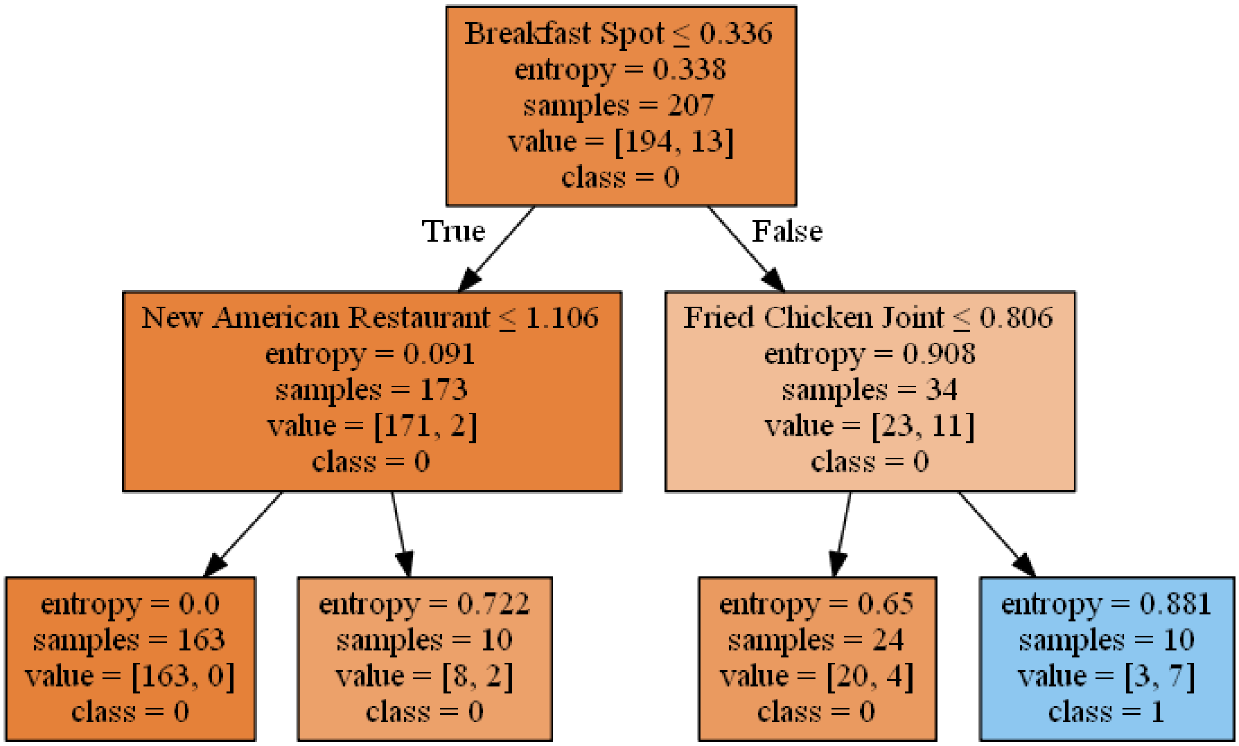


Figure 6: Tree Structure for a Decision Tree (Maximum Depth of 2) Predicting Single-Label Italian Restaurant Rating of Neighborhoods.

The tree obtained by fitting to single-label rating data first split neighborhoods based on prevalence of breakfast spots; after this the data was split further in one branch based on prevalence of New American Restaurants, and in split in the other branch by prevalence of Fried Chicken Joints. Only neighborhoods falling in the leaf corresponding to Breakfast Spots ≤ 0.398 and Fried Chicken Joints ≤ 0.806 were predicted to have a rating of 1 (two or more Italian Restaurants), with neighborhoods falling in all other leaves predicted to have a rating of 0 (zero or one Italian Restaurants).

C. Logistic Regression Classification

1. Multi-Label

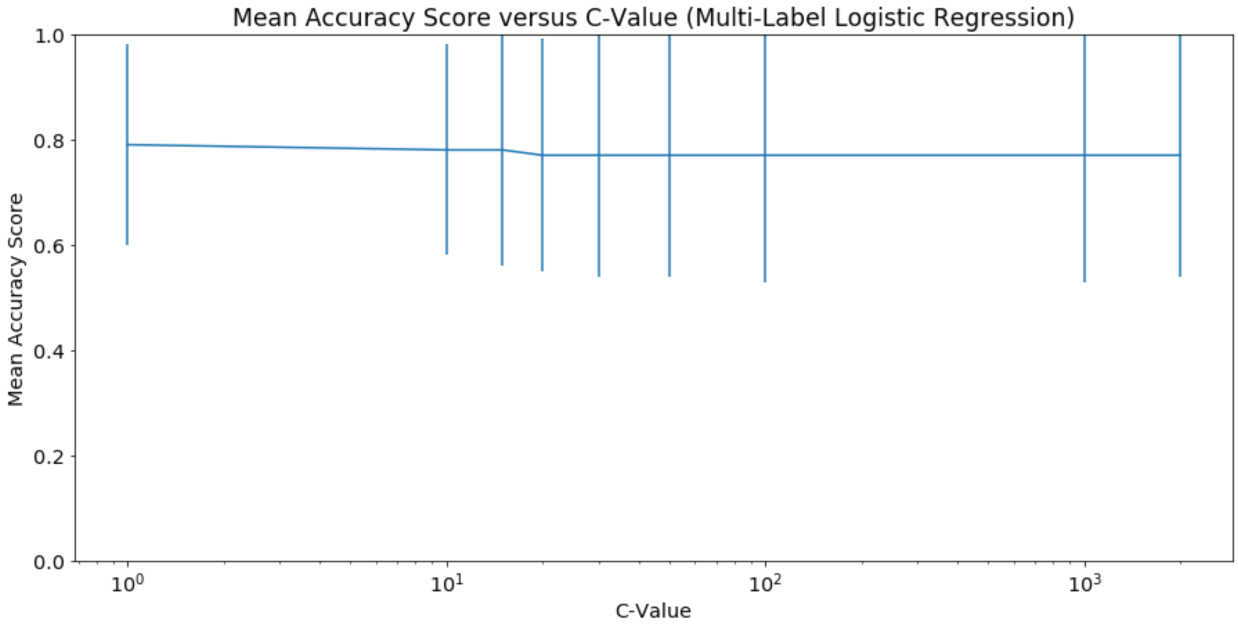


Figure 7: Accuracy Score Mean versus C-Value (Inverse Regularization Strength) for Multi-Label Logistic Regression Classifier. Error bars represent the standard deviation. Mean and standard deviation were calculated from 200 trials with random assignment of training and testing data.

Mean accuracy score for a Logistic Regression Classifier fit to multi-label data was highest at 0.79 for C value of 1, decreasing to 0.78 for C values between 10 and 15 inclusive and further to 0.77 for C values above 15. Standard deviation of the started at 0.19 for a C value of 1, increased with C value to peak at 0.24 for C values between 100 and 1000 inclusive, and dropped slightly to 0.23 for C values above 1000.

2. Single-Label

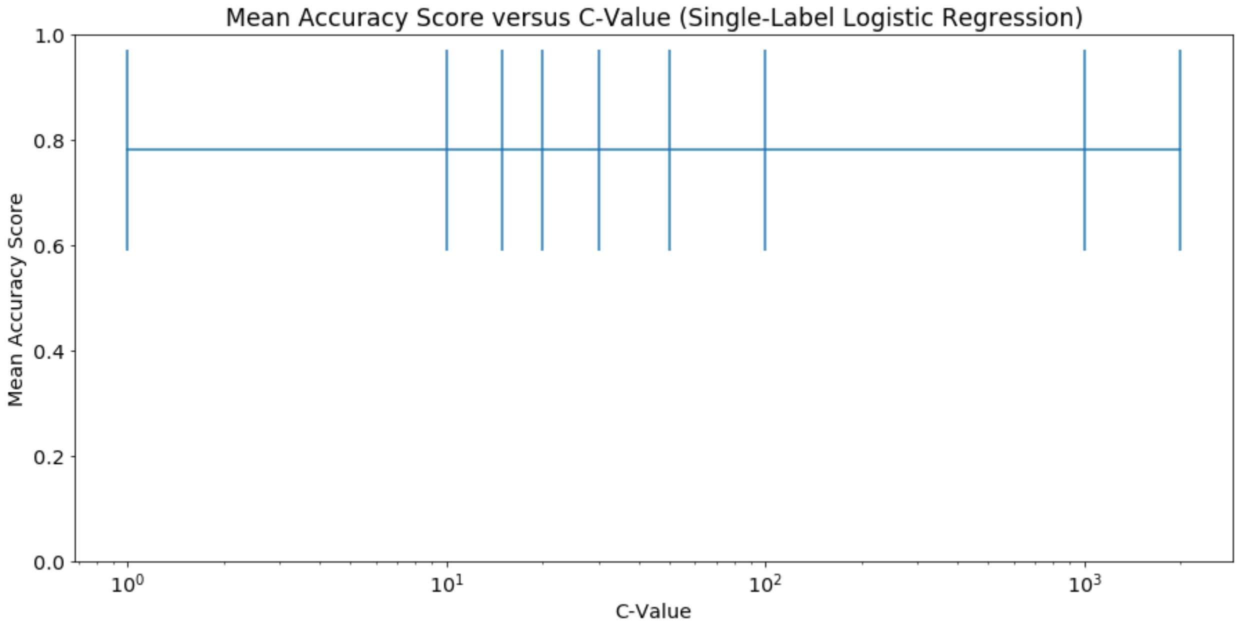


Figure 8: Accuracy Score Mean versus C-Value (Inverse Regularization Strength) for Logistic Regression Classifier fit to Single-Label Data. Error bars represent the standard deviation. Mean and standard deviation were calculated from 200 trials with random assignment of training and testing data.

For a logistic regression classifier fit to single-label data, the mean accuracy score was consistently 0.78, and standard deviation was consistently 0.19. Median accuracy score did show some variation, being 0.77 for C-values below 10 and 0.78 for C-values above 10.

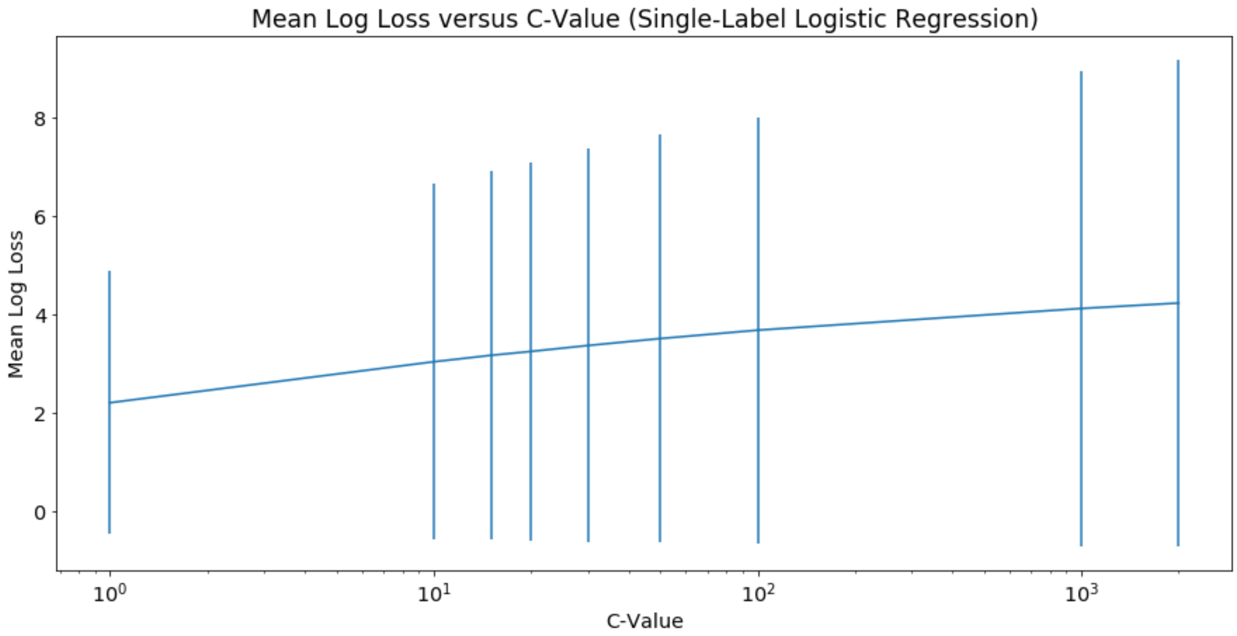


Figure 2: Log Loss Mean versus C-Value (Inverse Regularization Strength) for Logistic Regression Classifier fit to Single-Label Data. Error bars represent the standard deviation. Mean and standard deviation were calculated from 200 trials with random assignment of training and testing data.

The lowest mean log loss observed was 2.21, observed at a C-value of 1. Average log loss gradually increased with C-value, slowly leveling off; the highest log loss observed was 4.23. Standard deviation of log loss was also lowest at a C-value of 1, again increasing with C-value. Notably, the standard deviation of log loss was often consistently greater than the mean of log loss.

D. Overall Classification Summary

Table 1: Highest Mean Accuracy Scores by Model. Error indicates associated standard deviation. Mean and standard deviation calculated from 200 trials.

Classifier Model	Maximum Accuracy in Validation (Multi-Label)	Maximum Accuracy in Validation (Single Label)
K-Nearest Neighbors	0.77 ± 0.05	0.81 ± 0.05
Decision Tree	0.79 ± 0.05	0.93 ± 0.03
Logistic Regression	0.79 ± 0.19	0.78 ± 0.19

With the notable exception of Logistic Regression, the classifiers tended to produce higher accuracy when predicting single-label data compared to multi-label data. For multi

Multi-label logistic regression showed a standard deviation of 0.19 at a C-value of 1, which is also the value that produced the highest accuracy score of 0.79. A decision tree classifier with a depth of 1 also gave mean accuracy score of 0.79, but showed more reliability with a standard deviation of only 0.05; it therefore appears that the decision tree was the best option for multi-label classification.

The decision tree classifier also showed the best results for single-label classification, giving a mean accuracy score of 0.93 for a tree with one level and a standard deviation of only 0.03.

Looking More Closely at the Best Classifier

Table 2: Classification Support for Single-Label Decision Tree. Precision indicates rate of true positives to total positive guesses. Recall indicates rate of true positives to total actual positives. F1-Score is Harmonic Average of Precision and Recall. Support is number of cases used to calculate a parameter.

	Precision	**Recall**	**F1-Score**	**Support**
Group 0	0.97	0.98	0.98	294
Group 1	0.70	0.54	0.61	13
Micro Average	0.96	0.96	0.96	207
Macro Average	0.83	0.76	0.79	207
Weighted Average	0.95	0.96	0.95	207

A detailed classification report on a single-label decision tree classifier with maximum depth of 2 and fit to all the data shows it had nearly 99% precision and recall when a neighborhood had one or zero Italian restaurants. That is, when a neighborhood had less than two Italian restaurants, the classifier almost always correctly guessed the neighborhood rating, and when the classifier guessed a neighborhood had zero or one restaurants the guess was almost always correct. However, precision and recall were much lower for neighborhoods with more than one Italian restaurants.

Given that neighborhoods with one or zero Italian restaurants were so much more common, these results suggest that the classifier is heavily biased towards guessing '0 or 1 Italian restaurants', but appears fairly accurate because most neighborhoods do in fact fall in that category. If the model was applied to a group of neighborhoods where most had more than one Italian restaurant, it's accuracy would probably be much lower.

E. Neighborhood Group Comparisons

1. Counts

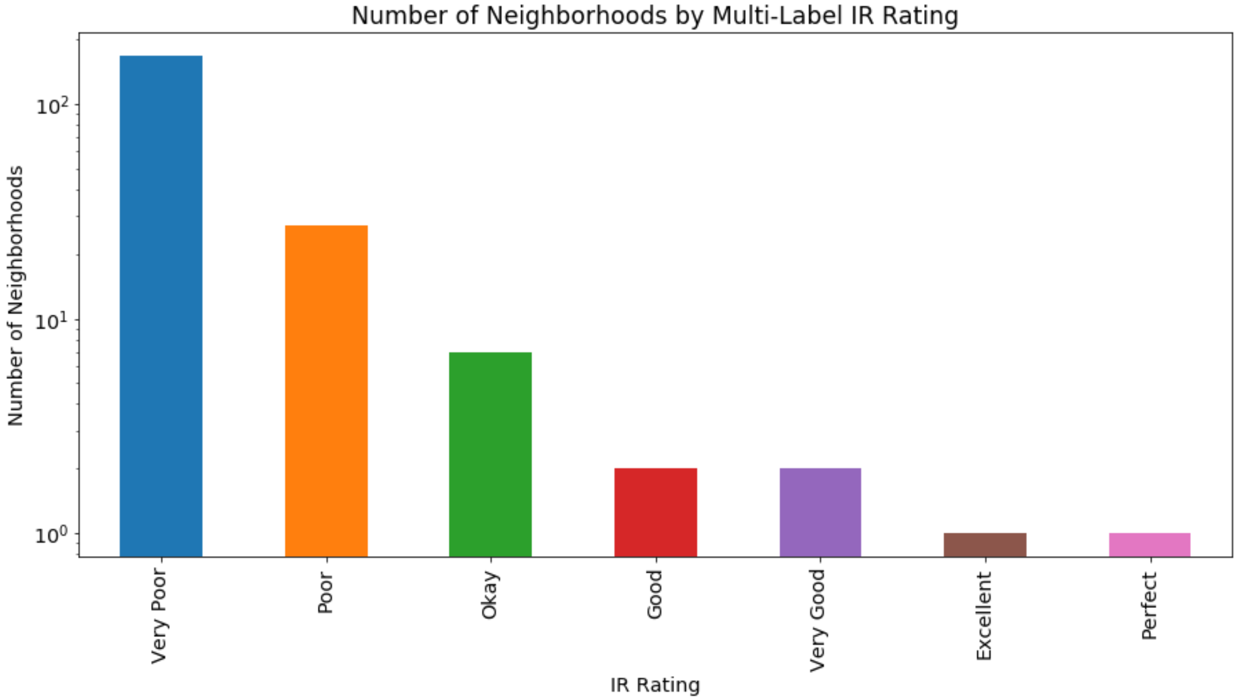


Figure 9: Number of Neighborhoods by Multi-Label IR (Italian Restaurant) Rating. 'Very Poor' indicates zero Italian restaurants per neighborhood, 'Poor' indicates one Italian restaurant, 'Okay' indicates two, 'Good' indicates three, 'Very Good' indicates four, 'Excellent' indicates five, and 'Perfect' indicates six Italian restaurants per neighborhood.

For the multi-label rating system, the vast majority of neighborhoods fell into the 'Very Poor' and 'Poor' categories. Seven neighborhoods had an 'Okay' rating (two Italian restaurants), the 'Good' and 'Very Good' ratings were assigned to two neighborhoods each, and the 'Excellent' and 'Perfect' ratings were assigned to one neighborhood each.

For the Boolean rating system, there were 194 neighborhoods with a rating of '0' (zero or one Italian restaurant), and 15 neighborhoods with a rating of '1' (more than one Italian Restaurant).

2. Correlations

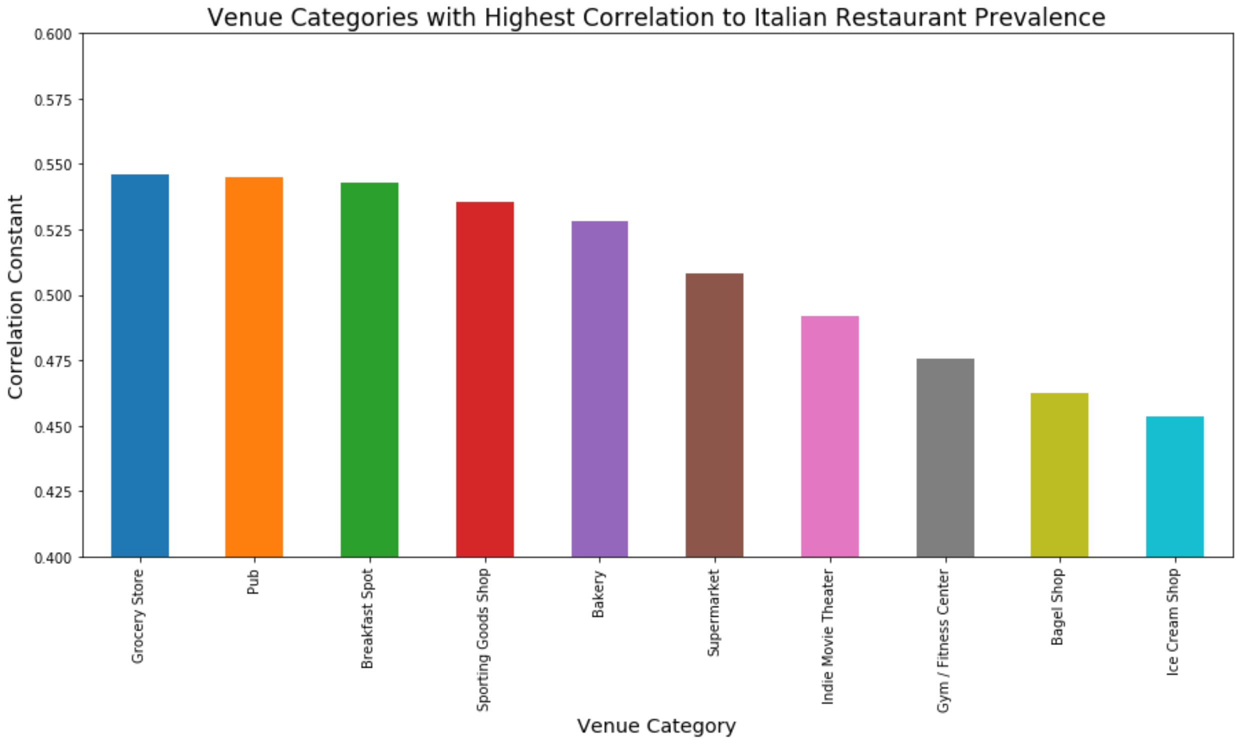


Figure 10: Highest Correlations in Prevalence between Italian Restaurants and other Venue Categories. Bar heights are calculated as Pearson's Correlation constant between vector of Italian restaurant counts per neighborhood and vector of counts per neighborhood for other venue category.

When examining correlation between prevalence of Italian Restaurants and prevalence of other venues, no venue had a negative correlation with Italian restaurants with absolute value > 0.058, whereas many venues had positive correlations > 0.1 with Italian restaurant prevalence. Thus, only the venues with the largest positive correlation values will be reported.

The ten venue categories whose prevalence correlated most strongly with that of Italian Restaurants were, from strongest to weakest correlations: Grocery Store, Pub, Breakfast Spot, Sporting Goods Shop, Bakery, Supermarket, Indie Movie Theater, Gym / Fitness Center, Bagel Shop, and Ice Cream Shop. The strongest correlation with Italian Restaurants observed was just under 0.55 for Grocery stores.

3. Comparing Neighborhoods

For simplicity, only neighborhoods divided by single-label Italian Restaurant Rating are compared.

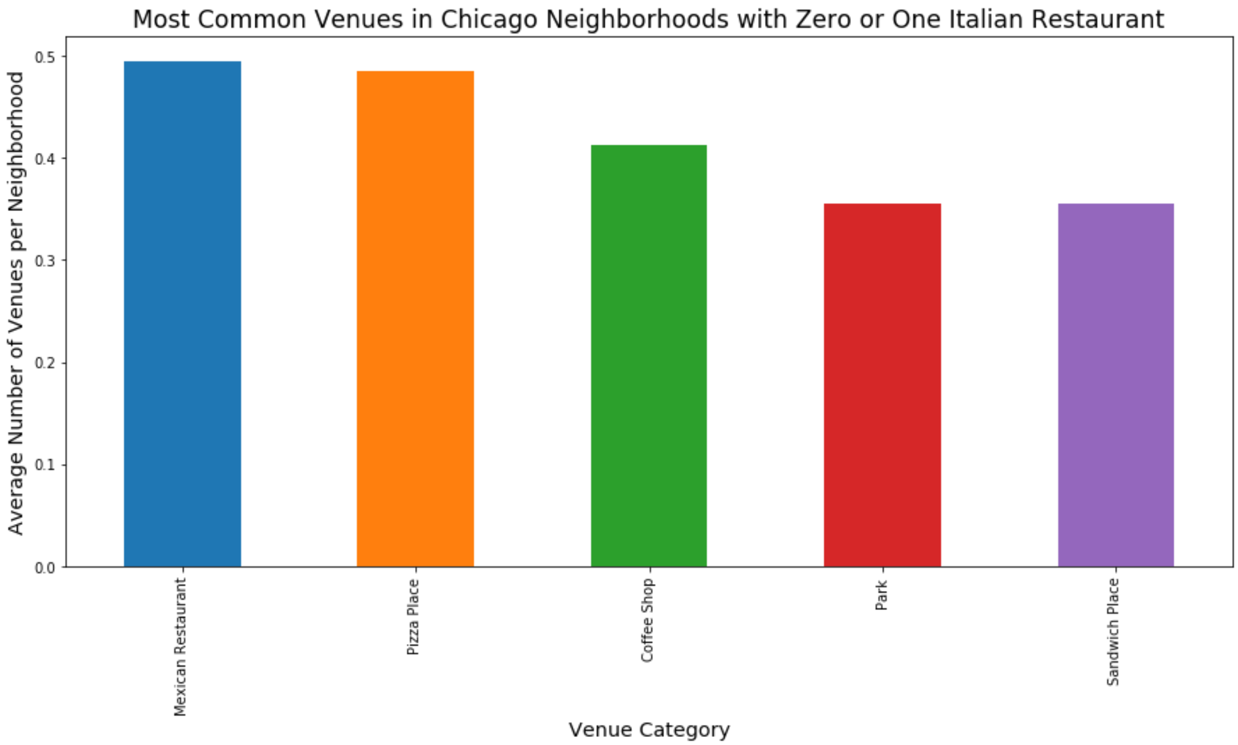


Figure 11: Top Five Most Common Venues in Chicago Neighborhoods with Zero or One Italian Restaurant. Bar heights represent number of venues per neighborhood, averaged across all Chicago neighborhoods with zero or one Italian restaurant.

For neighborhoods with zero or one Italian Restaurant, the five most common venue categories (from most common to least common) were Mexican Restaurant, Pizza Place, Coffee Shop, Park, and Sandwich Shop.

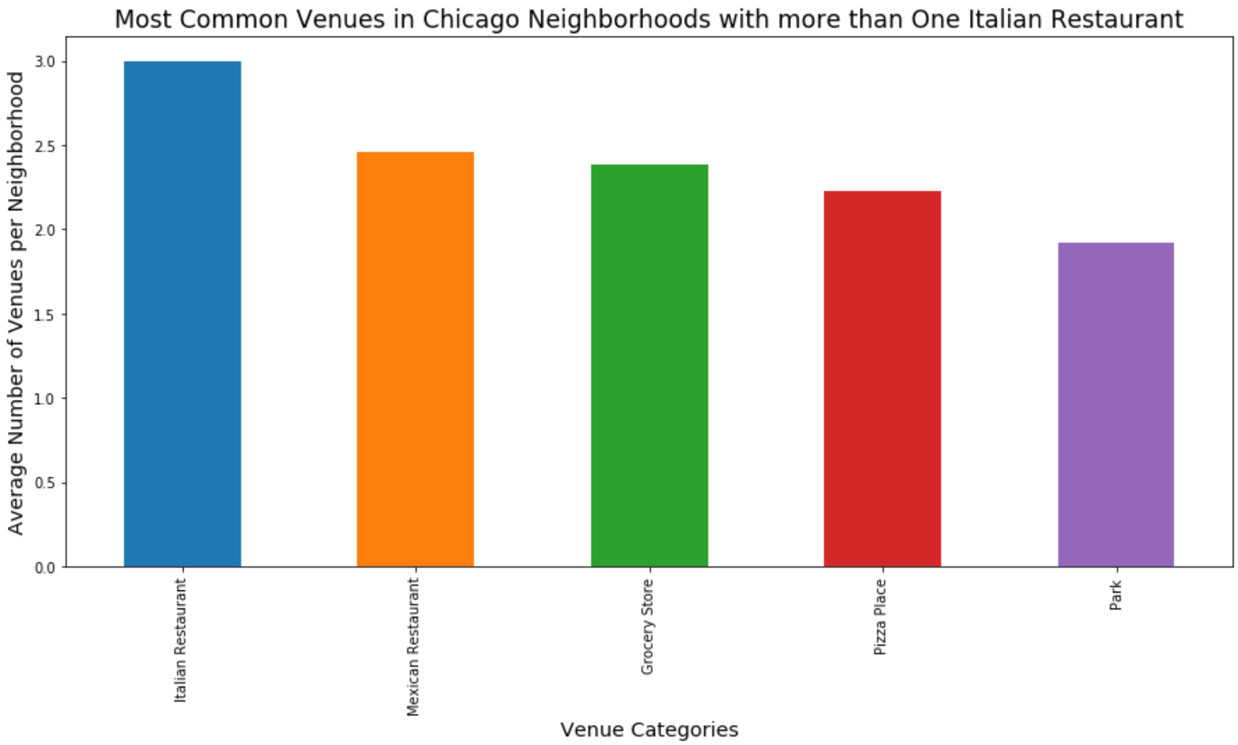


Figure 12: Top Five Most Common Venues in Chicago Neighborhoods with More than One Italian Restaurant. Bar heights represent number of venues per neighborhood, averaged across all Chicago neighborhoods with zero or one Italian restaurant.

For neighborhoods with more than one Italian Restaurant, the five most common venue categories, in order of descending prevalence, were Italian Restaurant, Mexican Restaurant, Grocery Store, Pizza Place, and Park.

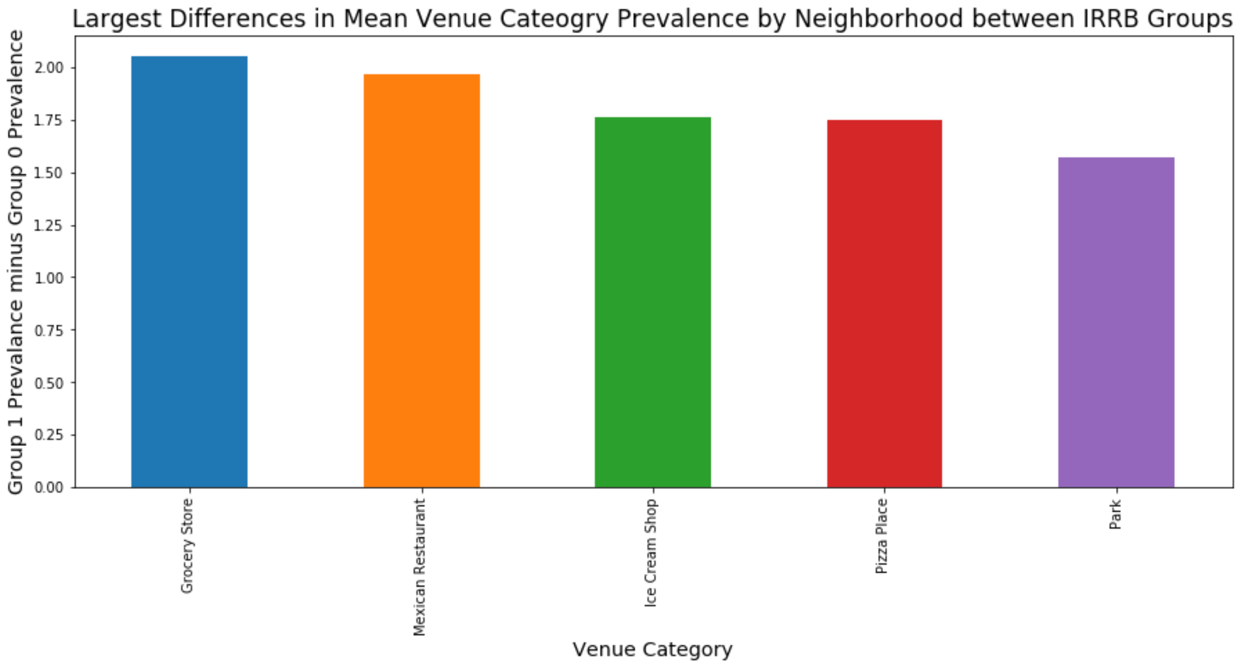


Figure 13: Difference in Venue Category Mean Count between Neighborhoods with Multiple Italian Restaurants and Neighborhoods with Zero-One Italian Restaurant. Only the five venue categories with the greatest difference are shown.

Looking at difference in mean venue count between neighborhoods with multiple Italian Restaurants and neighborhoods with zero or one Italian restaurants, the venue categories that showed the greatest difference (more common in neighborhoods with multiple Italian restaurants) were (in order of descending difference) Grocery Store, Mexican Restaurant, Ice Cream Shop, Pizza Place, and Park.

F. Mapping

Multi-Label Italian Restaurant Rating

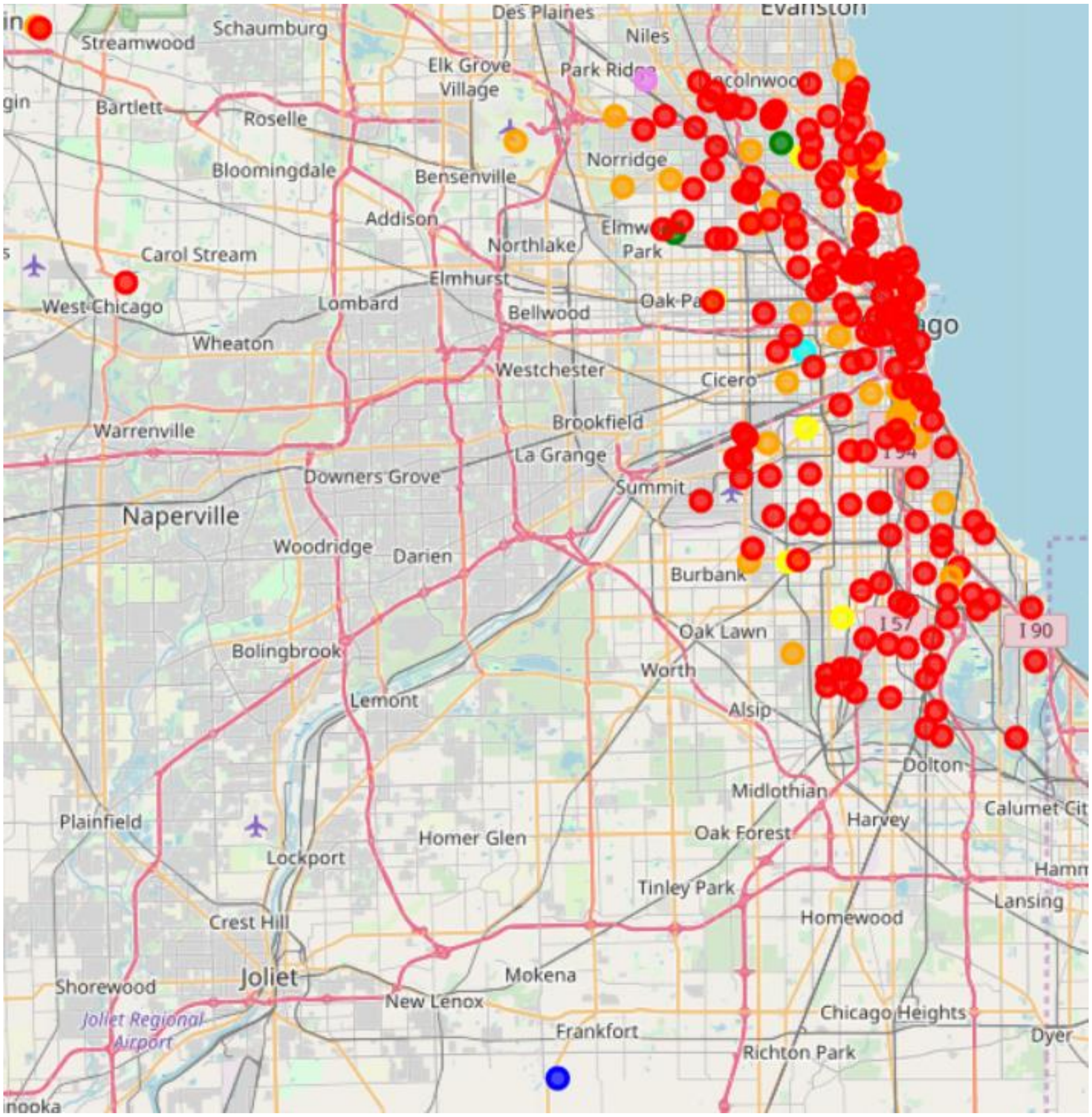


Figure 14: Mapping of Chicago Neighborhood Centers, Colored by Number of Italian Restaurants. Red indicates no Italian Restaurants, orange for one Italian Restaurant, yellow for two, green for three, cyan for four, blue for five, purple for six

Mapping the neighborhoods colored by rating does not reveal any obvious trends. It does look somewhat like neighborhoods with more than three Italian restaurants tend to occur along major roads. Note that out of the two 'Very Good' (cyan) neighborhoods, only one is visible; the other is covered up in the heart of the densest and largest cluster of red neighborhoods, in the upper right where the name 'Chicago' is partially visible.

V. Discussion

A. Classification Accuracy

Overall it appears that the decision tree classifier was the best approach for predicting the suitability of a neighborhood for Italian restaurants, especially when the suitability/rating target variable is encoded as a single-label categorical variable (Table 1).

A closer analysis of the best classifier showed it was highly accurate when a neighborhood actually had zero or one Italian restaurant, but not very accurate when a neighborhood had more than one Italian restaurant (Table 2). Given that neighborhoods with zero or one Italian restaurants were so much more common, these results suggest that the classifier is heavily biased towards guessing '0 or 1 Italian restaurants'; it appears fairly accurate overall only because neighborhoods with zero or one Italian restaurant are so much more common than those with more than one Italian Restaurant. If the model was applied to a group of neighborhoods where most had more than one Italian restaurant, it's accuracy would probably be much lower.

B. Decision Tree Structure

The most accurate classifier we produced was a two-level decision tree classifier fitted to the single-label data (Figure 6), discussed in the results section. The structure of the tree suggests that Italian restaurants are more common in neighborhoods that lack Breakfast Spots; perhaps a neighborhood with Breakfast Spots is less likely to have venues aimed at serving lunch and dinner, such as Italian restaurants. The tree also indicates that Italian Restaurants are more common in neighborhoods with at least one Chicken Joint; this might indicate the Italian Restaurants tend to co-occur with other small-to-medium restaurants and fast food joints, forming clusters such as might be found off a freeway or highway exit.

C. Neighborhood Group Comparison

Out of 207 neighborhoods, 194 had 0 or 1 Italian restaurant, leaving only 13 with more than one Italian restaurants.

Among the other venues, there appears to be some sort of relationship between Italian Restaurants per neighborhood and grocery stores per neighborhood. Grocery Store was the venue category with the highest correlation to Italian Restaurant in the venue count Data Frame (Figure 10). When comparing neighborhoods with zero or one Italian Restaurant to those with multiple Italian restaurants, the venue category showing the greatest difference in mean count per neighborhood was grocery stores (Figure 13); 'Grocery Store' was the third most common venue category on average in neighborhoods with more than one Italian Restaurant (Figure 12), but was not in the top five most common venue categories on average for neighborhoods with zero or one Italian Restaurants (Figure 11).

D. Mapping Neighborhoods by Rating

The map coloring neighborhoods by multi-label or single-label Italian Restaurant Rating didn't show many obvious trends, but a few observations might be notable. Many neighborhoods fell in one of two dense clusters up against the coastline of Lake Michigan; a smaller cluster roughly around Lincoln Park, and a larger cluster around where the word ‘Chicago’ is partially covered up on the map (figure 14). Out of all these neighborhoods, only one (University Village, ‘Very Good’ rating) has a rating above ‘Okay’.

More generally, University Village is the only Chicago neighborhood east of IL 1 with a multi-label rating above ‘Okay’. Looking at the single-label map, University Village and Belmont Central are the only neighborhoods east of IL 1 and North/South Halsted Street with more than one Italian restaurant.

Notably, the ‘Little Italy’ neighborhood has a ‘Very Poor’ rating with no Italian Restaurants, but that might just indicate that this is a mostly residential neighborhood with few restaurants in general.

The two best neighborhoods for number of Italian restaurants appear to be located at the outskirts of Chicago. Edison Park, with a rating of ‘Perfect’ (6 Italian restaurants) is found at the northern edge of the main cluster of Chicago neighborhoods. Big Oaks, with a rating of ‘Excellent’ (five Italian restaurants) is the southernmost of three or four ‘outlier’ neighborhoods whose centers lay outside the main cluster of neighborhoods.

VI. Conclusions

The goal of this project was to study the relationship between prevalence of Italian restaurants in Chicago Neighborhoods and prevalence of other venue categories in Chicago neighborhoods. It was hypothesized that the suitability of a neighborhood for Italian restaurants, measured by existing Italian restaurants, could be predicted by classification modeling with counts of other venue categories per neighborhood as predictors.

The results showed that looking at counts of other venue categories might give some idea of how suitable a neighborhood is for Italian Restaurants, but the accuracy of this method is questionable. Models attempting to predict the specific number of Italian restaurants had at best around 77-78% accuracy; this seems fairly high at first, but a ~25% error rate is probably unacceptable for a model being used for professional purposes. Models merely predicting whether or not a neighborhood had more than one Italian Restaurant showed much higher accuracy, but this appears to have been somewhat illusory; the best model was heavily biased towards predicting zero or one Italian restaurants per neighborhood, only seeming accurate because neighborhoods with more than one Italian restaurant were relatively rare.

Overall, while some interesting trends were noted, it appears that the approach taken was too simplistic to predict or explain prevalence of Italian Restaurants. There was some evidence that Italian restaurant prevalence per neighborhood could be related to numbers of grocery stores, Fired Chicken Joints, and Breakfast Spots, but this was far from enough to explain the observed variation. A more detailed analysis is called for, taking into account factors such as neighborhood crime rates, typical neighborhood income, income of nearby neighborhoods, neighborhood accessibility (by public transit, or by highways and freeways), neighborhood location relative to city center and city limits, etc.