Cracking the Restaurant Industry

Problem

Can we find what makes a restaurant successful in order to help prospective restaurant owners?

It is estimated that 60% of restaurants fail in the first year, while 80% fail in the first 5 years. The restaurant industry is therefore risky: the initial capital outlay is significant, the recurring fixed costs are high, the market is very competitive and customers are volatile. Therefore, it would be beneficial for prospective and current restaurant owners to identify which factors contribute to popularity.

Data set

Our data came from the 2017 Yelp Dataset Challenge, a proprietary dataset released by the review aggregation site Yelp. It contains 4.1 million reviews for 144,000 business, of which 48,000 of these are restaurants. This is supplemented by 947,000 'tips' from users as well as check-in times for most of the businesses listed. The data are provided as json trees which were converted to rdata format data frames for further processing. We choose to analyze restaurants in two cities because analyzing the entire dataset was too computationally expensive for the methods we wanted to use, and because there are significant geographical differences which affect restaurants in different areas. Toronto and Las Vegas had the highest numbers of restaurants so they were selected.

Variables

Each restaurant entry in the 'businesses' dataset contains the following variables: a unique business ID hash, name, neighborhood, address, city, state, post code, latitude, longitude, average star rating, number of reviews, open or closed state (1 or 0 - Yelp keeps data on restaurants that have closed down), an 'attributes' list, a 'categories' list and an 'hours of the week' list. We created two subsets, for restaurants in each city, by filtering by the category 'restaurants' and by each of the city names. We supplemented these new tables with the number of checkins for each business (calculated by parsing and summing the check-in times listed in the 'checkins' table), and expanded the 'attributes' and 'categories' lists into separate columns for each categorical variable included. One notable variable included in 'attributes' was an average price range integer, ranging from 1 to 4.

We define our success metric *performance = (stars + is_open) * ln(reviews + check-ins)*, where the variables are defined as above. This was selected in order to simplify analysis and provide a dependent variable that included as much information as possible and could be used as a proxy for the popularity of a business.

Methodology

The following tools were used in implementing our analysis: logistic regression; CART trees; k-means clustering; hierarchical clustering; and textual analysis.

Categories

We looked at the success of different categories of restaurant (e.g. 'steakhouse', 'Italian') in two different ways. First, we used hierarchical clustering to reduce the large number of categories to a manageable number of 'archetypes' of related categories, separating the restaurants into each archetype. The hierarchical method allows use of scree plots and dendrograms to choose the number of clusters and visualize them. With the restaurants clustered we then calculated average values of price range, average stars, number of reviews and check-ins, and performance, and compared these between archetypes. As a contrasting method to validate results from clustering, we used CART trees to split the group by significant categories that affected performance, testing different cp values until we had an interpretable model. Due to the nature of our recommendations, interpretability was more important to us than accuracy so we were not concerned with maximizing the R^2 value.

Attributes

In a similar manner to the categories above, we generated CART models by performance metric based on the attributes of each restaurant. These attributes included a broad range of variables, from objective ('has wifi', 'open 24 hours') to subjective ('good for dinner', 'quiet'). Since attributes are not related by 'closeness' in the same way categories are, we did not analyze them with clustering.

Location

To assess the impact of location within a city, we also clustered restaurants into 'districts' related by geographic closeness. Unlike for the categories, location data (latitude and longitude values) is numerical and continuous, so k-means clustering makes sense for this. Again, we used scree plots to determine the appropriate number of clusters for each city, and split the restaurants based on their location cluster. For each cluster we again computed average price, stars, reviews, check-ins and performance values. For further analysis, we took each of the top three performing districts in each city and ran categorical clustering (as above) on just the restaurants in that district, to compare how location impacts the type of restaurant that does best in that area.

Staying open

In contrast to our other methods that analyzed restaurants based on performance, here we wanted to purely look at factors that influenced a restaurant's likelihood to close down. With our 'is open' state as the dependent variable we created logistic regression models based on the list of categories, to determine which types of restaurant are riskiest to open in each location. We looked only at the significant factors that increased or reduced the odds of a restaurant staying open.

Reviews

As a final qualitative insight, though not used in our recommendations, we generated word clouds of the most frequent words that appeared in one- and five-star reviews in each city. This was done by creating a corpus from the full review text columns, standardizing (removing punctuation and numbers, converting to lowercase) and removing stopwords before plotting the word clouds.

Analysis and results

Due to the complexity of our analysis, it is important not to rely on the individual models used above, as each can only give some look into the trends that govern the success of a restaurant; they must be interpreted together to gain meaningful insight, which we do in the 'recommendations' section that proceeds.

Toronto

In looking at categories of restaurants for Toronto, hierarchical clustering (appendix 5, meta analysis in appendix 4) shows that archetypes we can define as 'brunch/bar' are most successful overall, followed by 'Japanese/sushi' and 'café/bakery'; these all have significantly higher performance scores than the other archetypes. Our CART category analysis (appendix 1) suggests a similar result, as it finds that brunch/comfort food places have the highest performance score, followed by bars (that also serve food) and then Japanese restaurants.

When we apply CART analysis to restaurant attributes in Toronto, we find that 'trendy', moderate noise level dinner restaurants score best, followed by 'hipster' places that are not rated good for lunch or dinner specifically. The CART tree picks out ambiance and noise level as the most important factors across all performance ranges, suggesting that it is wise for prospective restaurant owners to consider the atmosphere of their business carefully.

Looking at the location clusters for Toronto (appendix 3), as defined by k-means clustering (we select 8 districts), and then analyzing the most successful categories in each district, we find that the highest-performing districts, especially 2 and 7, also appear to be the densest (appendix 6). For both of these districts, further analysis shows that bars and cafés score highest. The next best performing cluster, 1, is slightly less dense suggesting it is just out of the town centre, and here Japanese and sushi restaurants also score well.

The results of our analysis of which restaurant types are most likely to fail (logistic regression; appendix 7), show that restaurants defined as 'lounges', as well as seafood, French and tapas restaurants all have higher odds of failing than any other restaurant category.

Las Vegas

Cluster analysis for Las Vegas (appendix 13, meta analysis in appendix 12) shows that the three most successful archetypes by performance score are 'bars/nightlife', 'cafés/sandwiches/brunch', and 'Japanese/Thai/Asian' restaurants, which all scored significantly higher than the next best. These results are closely mirrored in this case by the category CART analysis (appendix 9), which allocates the highest scores to bars, Thai restaurants and brunch places.

Our attribute analysis CART model (appendix 10) suggests that better performing restaurants are either termed 'good for dinner' with no other necessities, 'good for brunch' but only if also 'casual' in ambiance, and 'good for dessert' but only if also not 'casual' in ambiance. This suggests that it is important to consider the ambiance of a restaurant in relation to the time of day that most custom is expected and the customer demographics that are being targeted; in the general case a dinner restaurant appears to be the most versatile in Las Vegas.

The map of location clusters (appendix 11) highlights that the districts in the south and west outskirts, just away from the densest strip area, house the best performing restaurants (appendix 14). By running our hierarchical clustering to find the best categories for each cluster, we find that Asian/Japanese places do best in cluster 2, fast food and bars do best in cluster 7, and cafés and bars do best in cluster 5.

The logistic regression analysis of categories most likely to influence closure in Las Vegas (appendix 15) suggests that Italian restaurants (but not pizza, so this suggests higher-end restaurants only) have significantly higher odds of closing than any other category. However, breakfast/brunch and pubs (but not bars) are correlated with lower odds of closing.

Recommendations

Summing up our analyses, we find that in Toronto the most popular categories of restaurants are brunch/breakfast and grille/dinner places. This can be, in many cases, consistent with the most popular attributes: a 'trendy' or 'hipster' ambiance. In terms of locations, we recommend looking at district 1 for grille restaurants, and districts 2 and 7 for a brunch/breakfast places. We would discourage prospective restaurant owners from a opening a seafood or French restaurant, or a tapas bar, without serious further research.

Looking at Las Vegas, we see that like Toronto, brunch/breakfast places have been particularly popular among Yelp users, together with Japanese/Asian fusion restaurants and various types of bars. This recommendation is supported by the highest performing attributes being 'good place for breakfast', 'café', and 'good place for dinner'. We would suggest looking at district 2 for Asian restaurants, district 5 for cafés and bars, and avoiding the category Italian in general because it is likely to be a saturated market in this area.

Impact

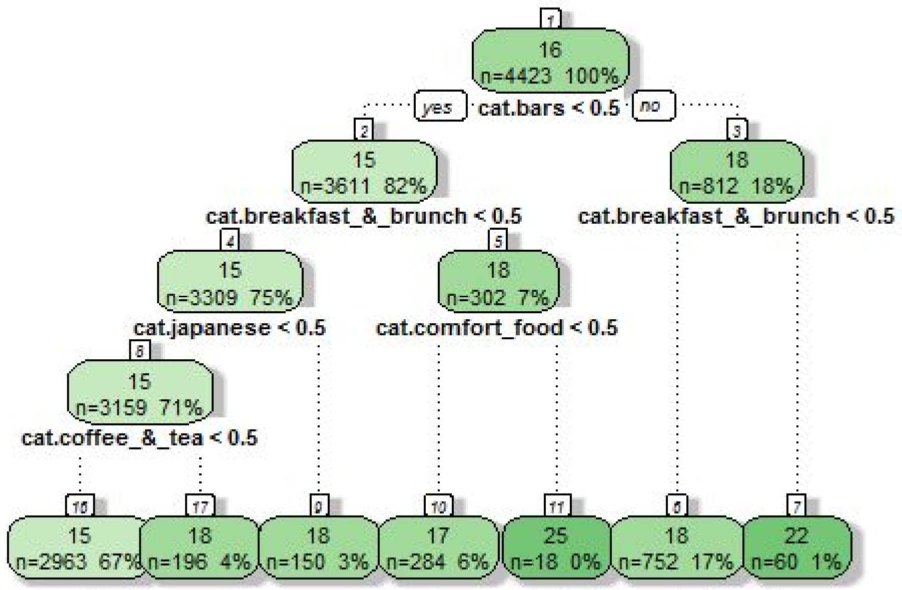
We are confident that our model provides significant value to prospective restaurant owners in Toronto and Las Vegas. First and foremost, it reveals which categories, attributes and locations yield the highest expected performance in terms of positive Yelp reviews. Users who leave reviews are likely to be influencers among their friend circle, so attracting praise from these users is likely to yield a disproportionate increase in business compared to other customers. Second, it indicates which categories of restaurants are likely to stay open, and which are not, which could be a beneficial model for risk-averse funding decisions such as bank loans. Third, for restaurant owners that already have a preference for a category, set of attributes or location - for instance, those who want to base the choice of cuisine on their own cultural background - our model can provide an estimation of expected popularity, and therefore success. It can also help choose the remaining independent variables by filling in various combinations of restaurant attributes and running these against our archetype cluster, CART, location and logistic models to estimate their potential performance.

Outlook

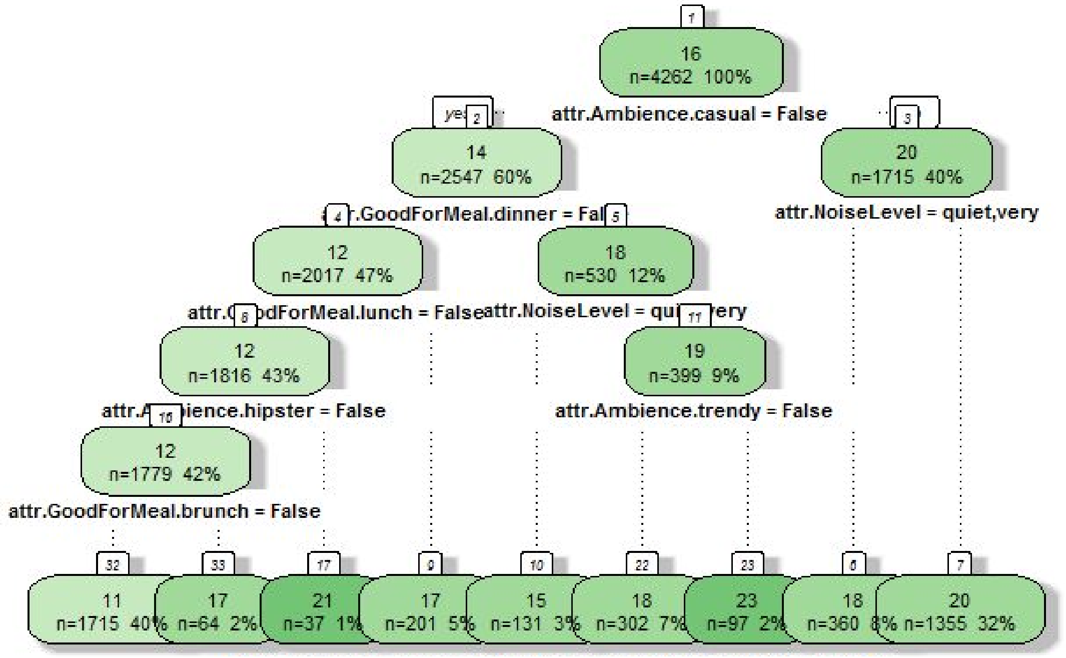
In this first version of our model, we provide mainly categorical guidance as we suggest which categories, attributes and locations to favour over others. In light of the significant upfront investments necessary, thin margins and uncertain time horizon of operations, it would be very valuable for prospective restaurant owners to get insight in actual dollar terms rather than a synthetic 'performance score'. Currently, our model treats all reviews and check-ins equally and just looks at quality and quantity. However, as restaurants typically have a target clientele, a future version of the model could offer a more fine-grained differentiation at the user level and assign different weights to the engagement of different user profiles. We could also make use of the 'price range' variable and use this, combined with number of checkins, to calculate a very rough estimate of yearly revenue.

The present version of our model only looks at the point in time when the data snapshot was pulled from the Yelp platform. Yet, as the restaurant landscape in metropolitan cities like Toronto and Las Vegas, as well as customer preference and tastes, are changing rapidly, it would be beneficial to have a 'live' model, which could be achieved by connecting to Yelp's API. This could look at variance in popularity of restaurants over time, and the effect that these surges have on their overall success.

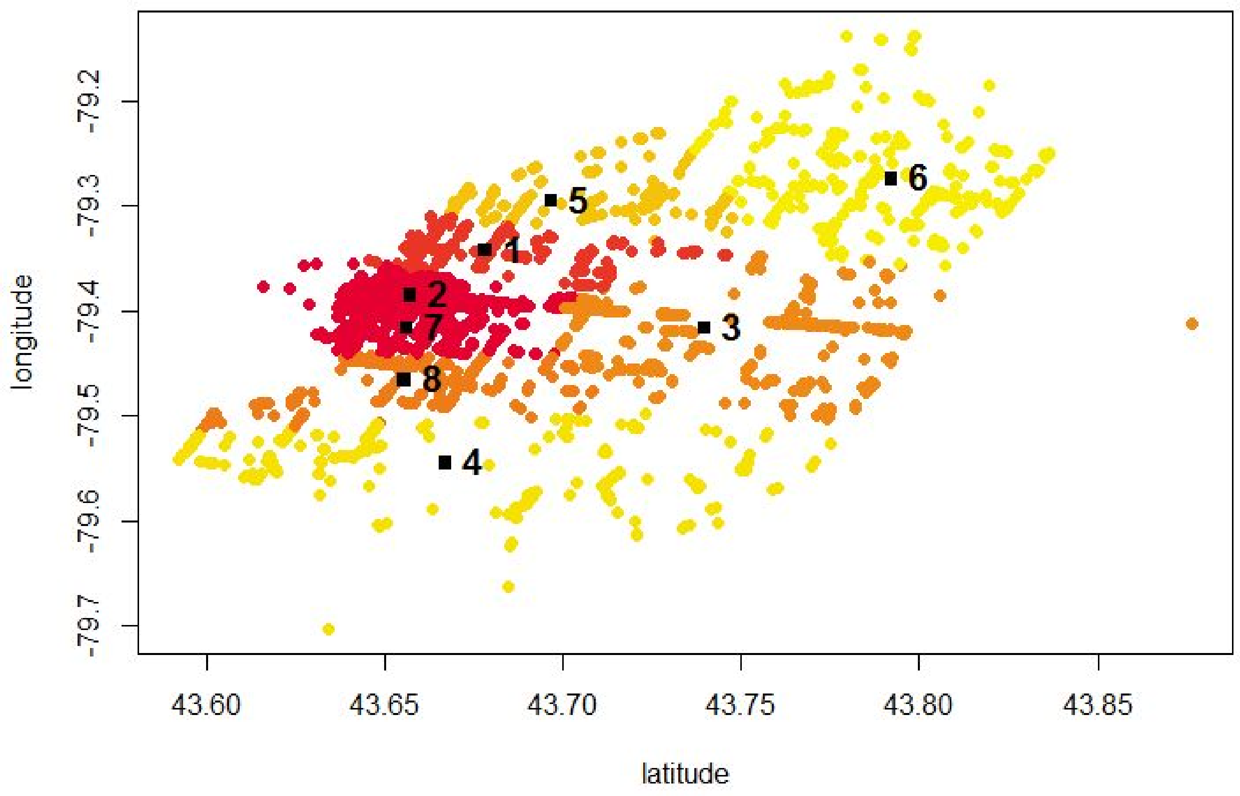
Appendices: Toronto



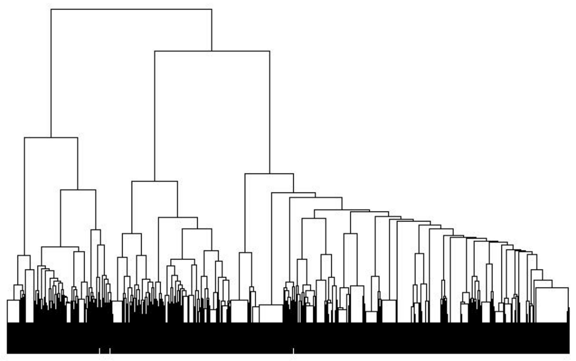
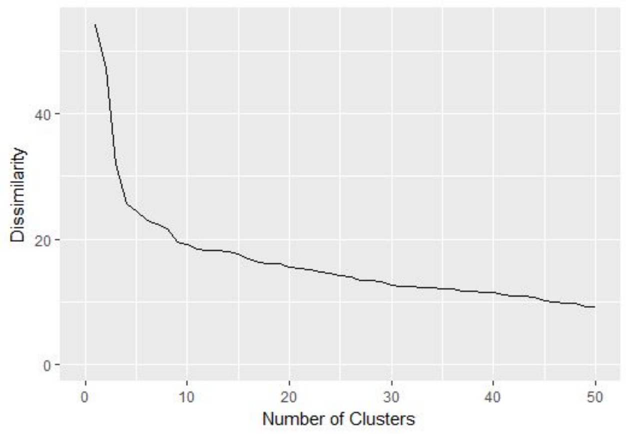
Appendix 1: Performance score by category - Toronto



Appendix 2: Performance score by attribute - Toronto



Appendix 3: Location clusters - Toronto

Appendix 4: Hierarchical cluster dendrogram (L) and scree plot (R) for category clusters - Toronto

	cats	price	stars	checkins	reviews	performance	n
1	pizza, italian	2.183946	3.481538	51.46769	39.14462	15.6092	325
2	fast_food, sandwiches, burgers, seafood, food, specialty_food	1.669281	3.350407	63.9709	36.84517	15.422	859
3	N/A	1.779427	3.412278	55.5916	36.56381	15.27083	3095
4	japanese, sushi_bars, bars	2.059211	3.489097	71.77259	46.29283	17.2874	321
5	nightlife, pubs, lounges, arts_._entertainment, bars, sports_bars	2.124343	3.389167	104.475	54.08167	18.03415	600
6	breakfast_._brunch, nightlife, pubs, bars, seafood, food, comfort_food, cafes, coffee_._tea, juice_bars_._smoothies, beer, wine_._spirits, wine_bars	2.02008	3.631179	93.45247	50.46768	18.44304	263
7	sandwiches, breakfast_._brunch, food, cafes, coffee_._tea, bakeries, desserts	1.586517	3.652807	70.37006	30.51351	17.215	481
8	chinese	1.65	3.137037	57.23333	30.16296	13.96808	270
9	pizza, italian, sandwiches, breakfast_._brunch, food, cafes, caterers, event_planning_._services, delis	1.975	3.541353	48.18045	31.4812	15.87334	133

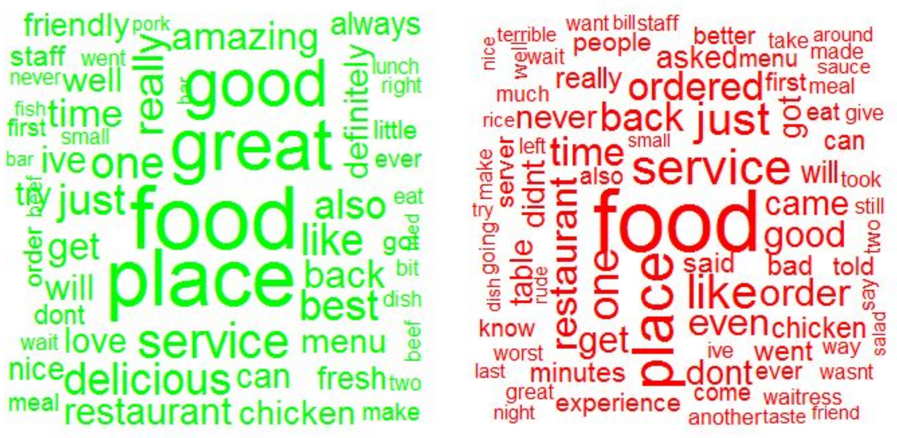
Appendix 5: Hierarchical archetype scores - Toronto

	price	stars	checkins	reviews	performance	n
1	1.840502	3.513333	54.63833	33.405	16.0683	600
2	1.863636	3.336315	88.43694	50.5644	16.75569	2236
3	1.932722	3.320225	52.07865	29.5014	14.94496	712
4	1.768473	3.516194	28.94737	16.7004	13.76858	247
5	1.736	3.585106	29.43617	19.17376	14.16383	282
6	1.622222	3.227571	42.49891	21.13786	13.62074	457
7	1.825703	3.564552	67.75075	44.51791	16.75601	1340
8	1.769932	3.575053	34.83721	25.7315	15.10916	473

Appendix 6: Location cluster scores - Toronto

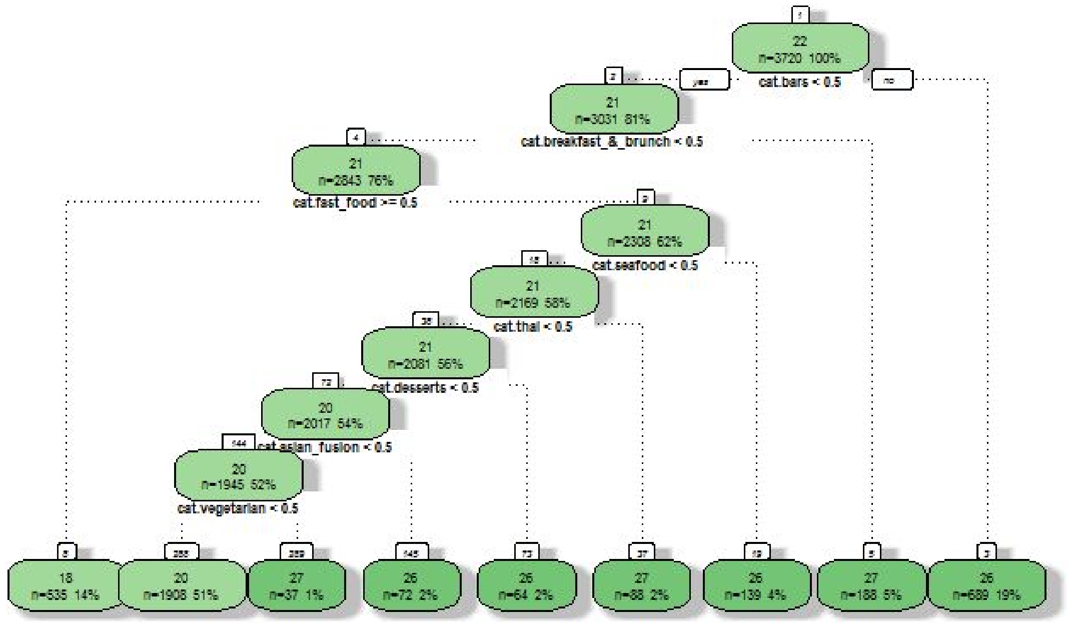
```
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -0.251253   0.182784  -1.375 0.169259
stars                            0.116691   0.049794   2.343 0.019106 *
review_count                     0.008287   0.002431   3.409 0.000653 ***
checkins                         0.005314   0.001409   3.772 0.000162 ***
cat.pizza                        0.682092   0.159383   4.280 1.87e-05 ***
cat.fast_food                    0.495492   0.223849   2.214 0.026862 *
cat.mexican                      0.559442   0.212694   2.630 0.008532 **
cat.korean                       0.418577   0.226787   1.846 0.064938 .
cat.lounges                     -0.714361   0.293846  -2.431 0.015054 *
cat.seafood                     -0.560874   0.252186  -2.224 0.026145 *
cat.sports_bars                  1.632189   0.625169   2.611 0.009033 **
cat.food                         0.437953   0.148390   2.951 0.003164 **
cat.salad                        1.080845   0.336654   3.211 0.001325 **
cat.greek                        0.542765   0.267161   2.032 0.042194 *
cat.cafes                        0.615688   0.174815   3.522 0.000428 ***
cat.chinese                      0.552069   0.147121   3.752 0.000175 ***
cat.french                      -0.880897   0.231971  -3.797 0.000146 ***
cat.halal                        0.678547   0.370552   1.831 0.067074 .
cat.caribbean                    0.439697   0.259138   1.697 0.089740 .
`cat.event_planning_&_services`  1.035241   0.563833   1.836 0.066346 .
cat.tapas_bars                  -0.765430   0.372911  -2.053 0.040113 *  ---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Appendix 7: Coefficients (only significant shown) in is_open logistic regression - Toronto

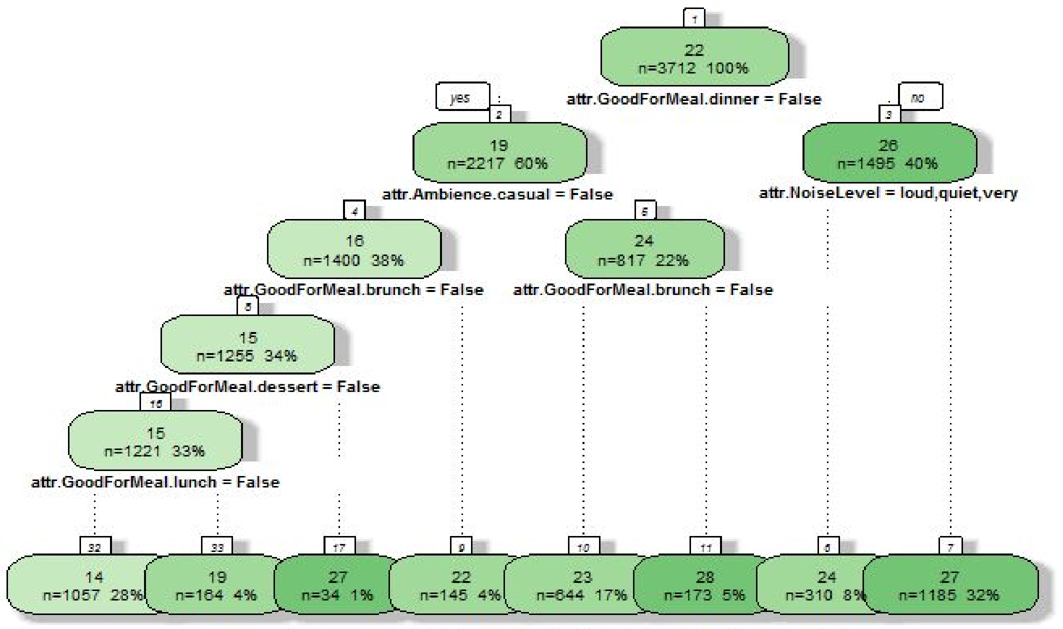


Appendix 8: Most frequent words in five-star (L) and one-star (R) reviews - Toronto

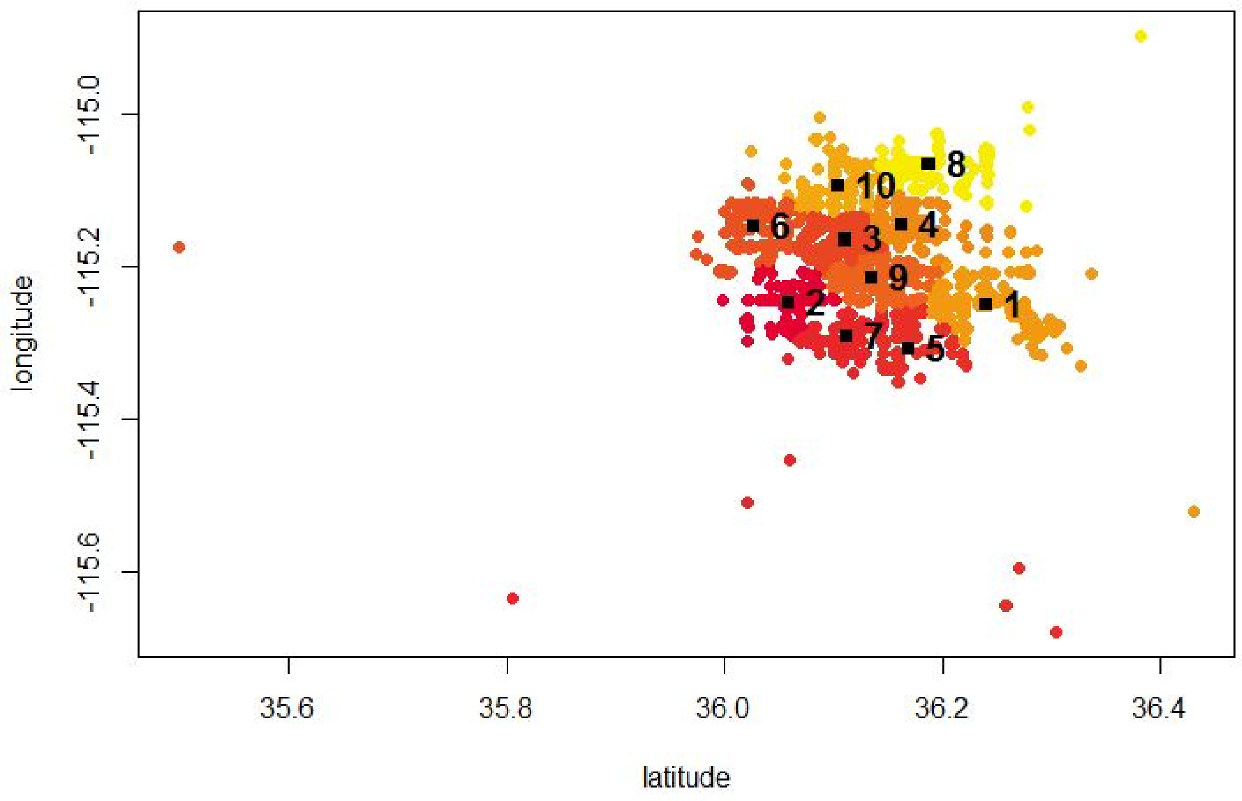
Appendices: Las Vegas



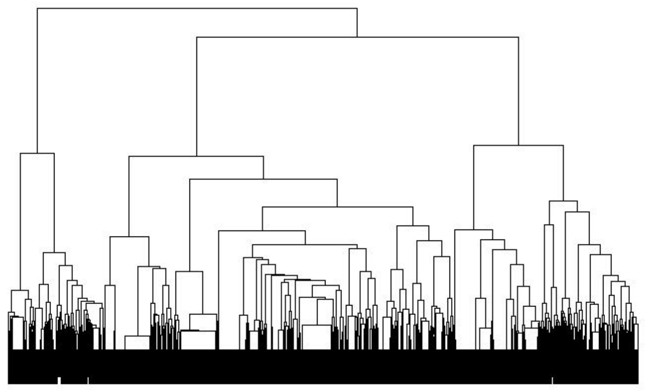
Appendix 9: Performance score by category - Las Vegas

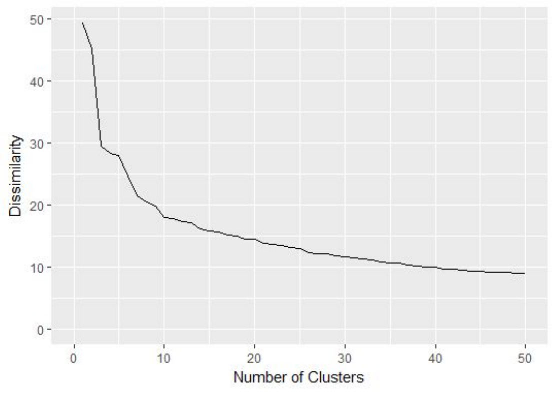

22
n=3712 100%
attr.GoodForMeal.dinner = False
yes
no
19
n=2217 60%
attr.Ambience.casual = False
26
n=1495 40%
attr.NoiseLevel = loud,quiet,very
16
n=1400 38%
attr.GoodForMeal.brunch = False
24
n=817 22%
attr.GoodForMeal.brunch = False
15
n=1255 34%
attr.GoodForMeal.dessert = False
15
n=1221 33%
attr.GoodForMeal.lunch = False
14
n=1057 28%
19
n=164 4%
27
n=34 1%
22
n=145 4%
23
n=644 17%
28
n=173 5%
24
n=310 8%
27
n=1185 32%

Appendix 10: Performance score by attribute - Las Vegas



Appendix 11: Location cluster - Las Vegas





Appendix 12: Hierarchical cluster dendrogram (L) and scree plot (R) for category clusters - Las Vegas

	cats	price	stars	checkins	reviews	performance	n
1	food, specialty_food, cafes, desserts, bakeries, mexican, seafood, coffee_._tea, event_planning_._services, caterers	1.690852	3.785501	556.281	148.5919	23.06972	669
2	chinese	1.758902	3.505293	552.2308	152.0445	21.4587	1417
3	food, nightlife, bars, sports_bars, lounges, pubs	2.034483	3.600833	809.6933	221.9017	25.57372	600
4	food, fast_food, burgers, chinese, mexican, sandwiches	1.127451	2.810976	249.897	40.42547	18.13907	738
5	food, chicken_wings, pizza, italian, sandwiches	1.683453	3.26431	275.0741	97.01515	18.65688	594
6	mexican, latin_american	1.344262	3.576623	278.0078	83.37922	20.13121	385
7	food, cafes, nightlife, bars, beer, wine_._spirits, vegan, sandwiches, coffee_._tea, breakfast_._brunch, juice_bars_._smoothies	1.703488	3.74	884.6743	178.1657	26.39071	175
8	food, burgers, sandwiches, delis, breakfast_._brunch, hot_dogs, diners	1.467446	3.441842	621.2052	171.9758	22.56201	619
9	food, bars, japanese, asian_fusion, sushi_bars	2.120172	3.817797	887.2161	239.3602	26.99633	236

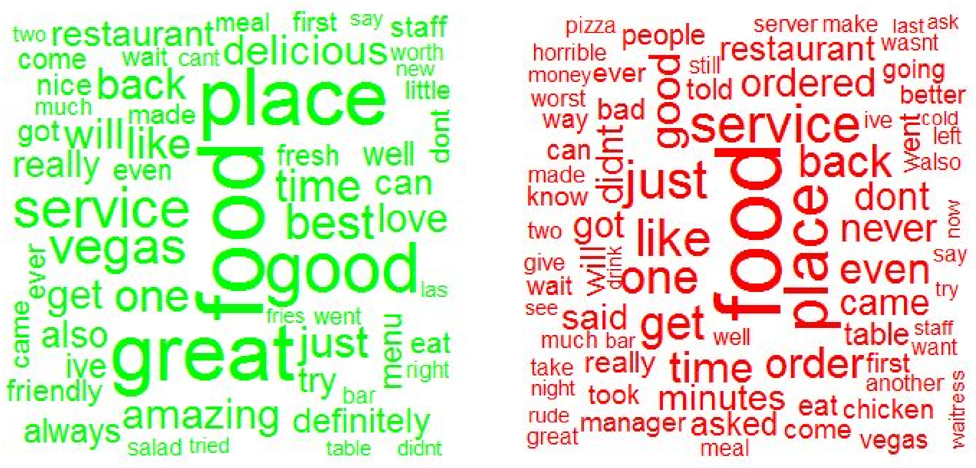
Appendix 13: Hierarchical archetype scores - Las Vegas

	price	stars	checkins	reviews	performance	n
1	1.445344	3.29717	272.9453	71.55849	19.91076	530
2	1.483553	3.512698	646.0571	129.5937	24.93893	315
3	1.988593	3.316875	800.9823	257.7104	22.65856	1357
4	1.604396	3.538593	407.5523	112.4151	20.43355	583
5	1.689759	3.438776	520.8892	129.8513	23.41444	343
6	1.50495	3.444181	475.1568	102.5914	22.28787	421
7	1.585366	3.66623	463.034	107.3586	23.71679	382
8	1.298969	3.284314	102.2353	27.70098	17.20469	204
9	1.538462	3.58587	559.7283	121.1435	21.7052	920
10	1.388889	3.490741	211.537	57.7672	19.41267	378

Appendix 14: Location cluster scores - Las Vegas

```
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                     1.420e-01  2.016e-01   0.704 0.481234
stars                          -1.096e-01  5.505e-02  -1.991 0.046477 *
review_count                    3.421e-03  7.090e-04   4.826 1.39e-06 ***
checkins                        4.571e-04  1.781e-04   2.567 0.010266 *
cat.food                        3.014e-01  1.493e-01   2.019 0.043527 *
cat.chicken_wings               1.024e+00  2.752e-01   3.720 0.000199 ***
cat.fast_food                   1.399e+00  1.990e-01   7.032 2.04e-12 ***
cat.pizza                       7.657e-01  1.441e-01   5.312 1.09e-07 ***
cat.chinese                     3.213e-01  1.538e-01   2.090 0.036632 *
cat.mexican                     5.954e-01  1.350e-01   4.410 1.04e-05 ***
cat.italian                    -4.916e-01  1.631e-01  -3.014 0.002581 **
cat.sandwiches                  2.916e-01  1.513e-01   1.927 0.053988 .
`cat.breakfast_&_brunch`        4.835e-01  1.990e-01   2.429 0.015134 *
cat.food_trucks                 1.099e+00  4.028e-01   2.729 0.006358 **
cat.salad                       1.033e+00  3.069e-01   3.367 0.000761 ***
`cat.event_planning_&_services` 1.353e+00  5.639e-01   2.399 0.016434 *
cat.pubs                        8.883e-01  3.517e-01   2.526 0.011535 *
cat.buffets                     5.848e-01  3.259e-01   1.794 0.072776 .
cat.diners                      6.304e-01  3.589e-01   1.756 0.079036 .
cat.latin_american              1.049e+00  3.888e-01   2.698 0.006980 ** ---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Appendix 15: Coefficients (only significant shown) in is_open logistic regression - Las Vegas



Appendix 16: Most frequent words in five-star (L) and one-star (R) reviews - Las Vegas

Appendices: Code

```
CalculateAndAddMetrics = function(x) {
  x$checkins = apply(x, 1, function(y) checkins$time[which(y$business_id == checkins$business_id)])
  rgx = "(?<=:)[0-9]+"
  lt = str_extract_all(x$checkins, rgx)
  x$checkins = NULL
  x['checkins'] = as.data.frame(unlist(lapply(lt, function(row) as.integer(sum(strtoi(unlist(row)))))))
  x$performance = (x$stars + x$is_open) * log(x$review_count+x$checkins)
  return(x)
}
ExpandCategoriesIntoFactorVariables = function(x) {
  categories = unique(unlist(as.list(x$categories),recursive=F))
  categories = categories[categories != "Restaurants"]
  categories = tolower(str_replace_all(categories, " ", "_"))
  categories = paste("cat.", categories, sep="")
  x$categories = tolower(str_replace_all(x$categories, " ", "_"))
  for (category in categories){
    x[category] = as.numeric(grepl(strsplit(category, "\\.")[[1]][2], x$categories))
    if (sum(x[category] == TRUE) < 0.01*nrow(x)) x[category] = NULL
  }
  x$categories = NULL
  return(x)
}
ExpandAttributesIntoFactorVariables = function(x) {
  testat =  unique(unlist(as.list(x$attributes),recursive=F))
  for (attribute in testat){
    if(length(grep("\\{", attribute)) != 0){
      name = strsplit(strsplit(attribute, "\\}")[[1]][1], "\\:")[[1]][1]
      subvalues = strsplit(strsplit(strsplit(attribute, "\\}")[[1]][1], "\\{")[[1]][2], ",")[[1]]
      for (subvalue in subvalues){
        subvalue = gsub(' ', '', subvalue)
        subvalue = noquote(strsplit(subvalue, "\\:")[[1]][1])
        subvalue = gsub('^.|.$', '', subvalue)
        coltitle = paste("attr.", name, ".", noquote(subvalue), sep="")
        if(!(coltitle %in% colnames(x))){
          regexpat = paste("(?<=", subvalue, "\\':\\W)[A-Za-z0-9]+", sep="")
          newvec = str_extract(x$attributes, regexpat)
          if (!all(na.omit(newvec)[1] == na.omit(newvec))) { x[coltitle] = newvec }
          regexdel = paste(attr, "\\':\\W[A-Za-z0-9]+", sep="")
          x$attributes = gsub(regexdel, "", x$attributes)
        }
      }
    } else {
      attr = strsplit(attribute, "\\:")[[1]][1]
      regexpat = paste("(?<=", attr, ":\\W)[A-Za-z0-9]+", sep="")
      coltitle = paste("attr.", attr, sep="")
      if(!(coltitle %in% colnames(x))){
        newvec = str_extract(x$attributes, regexpat)
        if (!all(na.omit(newvec)[1] == na.omit(newvec))) { x[coltitle] = newvec }
        regexdel = paste(attr, ":\\W[A-Za-z0-9]+", sep="")
        x$attributes = gsub(regexdel, "", x$attributes)
      }
    }
  }
  x$attributes = NULL
  return(averages)
}
RunLogisticAnalysis = function(x) {
  x$attributes = NULL
  x$categories = NULL
  x = subset(x, select=-c(type, hours, state, city, business_id, name, neighborhood, address, postal_code, longitude, latitude,
performance))
  split = sample.split(x, SplitRatio = 0.7)
  train = x[split == TRUE,]
  test = x[split == FALSE,]
  fit = glm(is_open ~ ., data=train, family="binomial")
  print(summary(fit))
  return(fit)
}
PlotWordCloud = function(x, color="black") {
  rc <- Corpus(VectorSource(x$text))
  rc <- tm_map(rc, removePunctuation)
  rc <- tm_map(rc, tolower)
  rc <- tm_map(rc, removeWords, c('the', 'this', stopwords('english')))
  wordcloud(rc, max.words=100, random.order=FALSE, colors=color)
}
```

```
RunCartAnalysis = function(x, cex=NULL, cp=NULL, sub="") {
  x$categories = NULL
  x$attributes = NULL
  x = subset(x, select=-c(type, hours, state, city, business_id, name, neighborhood, address, postal_code, longitude, latitude,
is_open, review_count, stars, checkins))
  split = sample.split(x, SplitRatio = 0.7)
  train = x[split == TRUE,]
  test = x[split == FALSE,]
  tree = rpart(performance ~ ., data=train, method="anova", cp=cp)
  fancyRpartPlot(tree, sub=sub, cex=cex)
}
RunLocationClustering = function(x, n=10) {
  x = ExpandAttributes(x)
  xLatLong = x[c('latitude', 'longitude')]
  km <- kmeans(xLatLong, iter.max=100, n)
  kmcluster = as.data.frame(km$cluster)
  kmcluster$col = character(nrow(kmcluster))
  x$cluster = kmcluster
  c1 = x[x$cluster==1,]
  clusters = list()
  clusters[0] = x[x$cluster==1,]
  averages = data.frame(price=numeric(n), stars=numeric(n), checkins=numeric(n), reviews=numeric(n), performance=numeric(n),
n=numeric(n))
  for (c in seq(1,n)) {
    cluster = x[x$cluster==c,]
    averages$n[c] = nrow(cluster)
    averages$stars[c] = mean(cluster$stars)
    averages$checkins[c] = mean(cluster$checkins)
    averages$performance[c] = mean(cluster$performance)
    averages$reviews[c] = mean(cluster$review_count)
    averages$price[c] = mean(as.numeric(cluster$attr.RestaurantsPriceRange2), na.rm=TRUE)
  }
  fcol <- colorRamp(c("#F8EC00","#E60031"))
  norms <- with(averages, (performance - min(performance)) / diff(range(performance)))
  for (c in seq(1,n)) { kmcluster[kmcluster$`km$cluster` == c,]$col = rgb(fcol(norms), maxColorValue=256)[c] }
  plot(xLatLong,col=kmcluster$col, pch=19)
  points(km$centers, col="black", pch = 15)
  text(km$centers, pos=4, cex=1.3, font=2)
  print(averages)
  return (x)
}
CreateCategoryClusters = function(x, n=9) {
  y <- ExpandCategories(x)
  x <- ExpandAttributes(x)
  y <- subset(y, select=-c(type, hours, state, city, business_id, name, neighborhood, address, postal_code, longitude, latitude,
stars, review_count, is_open, checkins, performance, attributes))
  d <- dist(y)
  mod.hclust <- hclust(d, method="ward.D2")
  cx <- data.frame(k = seq_along(mod.hclust$height),dissimilarity = rev(mod.hclust$height))
  ggplot(cx, aes(x=k, y=dissimilarity)) + geom_line()+xlab("Number of Clusters") +ylab("Dissimilarity") + xlim(0,50)
  assignments <- cutree(mod.hclust, n)
  y$cluster = assignments
  clustermeans = subset(data.frame(y[1,]), select=-c(cluster))
  clustercats = list(n)
  clusterscores = data.frame(cats=character(n), price=numeric(n), stars=numeric(n), checkins=numeric(n), reviews=numeric(n),
performance=numeric(n), n=numeric(n), stringsAsFactors=FALSE)
  for (c in seq(1,n)) {
    cluster = subset(y[y$cluster==c,], select=-c(cluster))
    clustermeans[c,] = colMeans(cluster)
    clustercats[c] = list(rownames(as.data.frame(which(apply(clustermeans[c,], 2, function(i) i>0.1)))))
    clusterx = x[rownames(cluster),]
    clusterscores$n[c] = nrow(cluster)
    clusterscores$stars[c] = mean(clusterx$stars)
    clusterscores$checkins[c] = mean(clusterx$checkins)
    clusterscores$performance[c] = mean(clusterx$performance)
    clusterscores$reviews[c] = mean(clusterx$review_count)
    clusterscores$price[c] = mean(as.numeric(clusterx$attr.RestaurantsPriceRange2), na.rm=TRUE)
    clusterscores$cats[c] =  paste(unlist(clustercats[c]), collapse=', ')[1]
  }
  return (clusterscores)
}
```

Appendix 17: R Code (main functions only)