Zomato (Bengaluru) Data Set Analysis

Daniel Sanchez
Prepared for the Master of Science in Business Analytics Degree Program,
University of North Carolina at Wilmington

BAN 530: Applications of Business Analytics
Dr. Rebecca Scott
Module 7 - Final Write-Up with Decision Analysis
August 15th, 2021

A. Project Scope

Executive Summary

Any new business venture requires a solid understanding of the market landscape. Success often hinges upon finding a niche or offering a service that consumers are currently lacking. The restaurant industry is no exception. Data suggests that the restaurant scene in Bengaluru, India contains untapped market potential. This report will reveal how a business can best configure itself to enter this competitive arena.

Data Set and Analysis Description

Two questions will drive this analysis:

1.) What restaurant features are most likely to be critical to the success of a new restaurant entering the Bengaluru market?

2.) How should a new restaurant structure its offerings and location(s) in order to compete?

In order to answer these questions, it will be important to better understand the data set used in our analysis. The source data comes from Zomato, a restaurant review-aggregation website. This information comes available as a .CSV file, comprised of 51,717 rows of 17 variables. The data set features are listed below:

URL	(Website Address)
Address	(Physical Address of Restaurant)
Name	(Restaurant Name)
Online_order	(Yes/No, if available option)
Book_table	(Yes/No) if available option
Rate	(Aggregate Rating on a 5-Point Scale)
Votes	(Popularity)
Phone	(Restaurant Phone Number(s))
Location	(Area of Bengaluru)
Rest_type	(Type of Restaurant)
Dish_liked	(Dish Most-Often Favored)
Cuisines	(Type of Cuisine Offered)
Approx_cost	(Approximate Cost for Two People)
Reviews_list	(Text of Written Reviews)
Menu_item	(List of Menu Items)
Listed_in(type)	(Additional Restaurant Type Description)
Listed_in(city)	(Additional Specification of Location)

Figure A.1

The variables in this data set will be analyzed in a variety of different ways. This report will first include a descriptive analysis of summary statistics to capture a broad range of overall trends and stratified relationships between variables. We will examine traits of top performers by area, using ratings and votes as variables of interest. We will also review how restaurant features vary by location, in terms of cuisine, cost, the availability of booking reservations, and ordering online. This will help to establish a general assessment of the restaurant market in Bengaluru.

This report will also include a predictive analysis component which will investigate the influence of items, such as operational structure (e.g., Location, Cuisine, Cost, etc.) on customer satisfaction. Understanding the factors that drive popularity and high ratings can help inform business decisions for a new market participant.

After exploring predictive elements of the data set, we will apply a prescriptive approach to evaluate appropriate pricing. This analysis will consider a series of proposed restaurant parameters. Taking into consideration market constraints and projections will support offering recommendations for approximate pricing.

Limitations

It is worth noting the limitations of our data set and the work not included in the scope of this project. While the restaurant data supplied reflects product offerings and customer sentiment, there is no indication of operating expenses or profitability. While ensuring a sustainable, profitable business model is very important, this analysis will not be included. Another limitation of this data set is that it is cross-sectional, containing a snapshot of aggregated features. There is no variable to show the age or rise in popularity of Bengaluru restaurants. Without seeing time-series trends in the market, it would be necessary to occasionally re-evaluate a restaurant's performance.

Final Deliverable

The analysis in this report will generate recommendations for a new restaurant entering the Bengaluru restaurant market. These recommendations will include quantitative measures that a business can apply, in order to succeed with a data-driven plan. While this analysis is specifically focusing on the Bengaluru restaurant market, a similar analysis may be applied to other cities and population centers.

B. Supplemental Notes on Methods

The analysis performed for this report will incorporate a progressive sequence of five methods:

1. Data Cleaning - In order to properly review and make sense of key findings, the data set first must be cleaned. This process will involve identifying and removing features of the data set that are unlikely to inform the next series of methods. Some variables may be duplicative or irrelevant in nature and could potentially be eliminated. The data set will also be inspected for rows containing erroneous information and null values. While some missing entries might be imputed, a high prevalence of irregular data may simply justify a row-wise deletion.
2. Descriptive Analysis (Summary Statistics) - Once the data set has been cleaned and normalized, this next phase will consist of broadly reviewing the data for summary statistics. The variables remaining in the cleansed data set are likely to contain informative insights into the overall restaurant market in Bengaluru. This phase of the analysis will be focused on the characteristics of the entire data set, as opposed to examining relationships between variables.
3. Exploratory Data Analysis (including Visualization) - This method will provide more insight into the relationships between restaurant features. A string of bivariate relationships will be studied, with results expressed using a variety of data visualization techniques (e.g., scatter, line, and bar plots).
4. Regression Analysis - This method will delve more deeply into the relationships between variables. Specifically, we will determine the degree of correlation between each data set feature and the response variables (Rate and Votes). We will also assess variable importance and investigate any apparent multi-collinearity and interaction between variables.
5. Prescriptive Analysis - Relying on information gleaned from the earlier analysis of summary statistics, this final method will maximize an objective function for approval ratings by outlining recommendations for key restaurant features. This will involve the application of market-derived constraints and additional assumptions for input values.

C. Data Cleaning and Understanding

The first steps of this analysis will be to examine the data set, prepare it for further review, and account for overall traits and characteristics. This will be conducted using the R programming language, Microsoft Excel, and Tableau. The Zomato data file can be retrieved from this page: https://bit.ly/3iLCtmh.

The following packages in R should be installed and loaded:

```
install.packages(tidyverse)
install.packages(VIM)
install.packages(tm)
install.packages(SnowballC)
install.packages(wordcloud)
install.packages(RColorBrewer)

library(tidyverse)
library(VIM)
library(tm)
library(SnowballC)
library(wordcloud)
library(RColorBrewer)
```

The document has been saved as a CSV file in our working directory. This document is imported into R and stored in an object, zomato. Once the file is imported, we can examine the structure of zomato (confirming 51,717 rows, spread across 17 columns). Using the *summary()* function, we can review the data types of the variables. Most of these variables come through as character types, aside from the numeric-type variables, “approx_cost(for two people)” and “votes.” We can then use the *glimpse()* function to visually inspect the first entries in all columns. This process reveals that some features (such as “online_order” and “book_table”) should potentially be reclassified as multi-level factors, while others should be reclassified to better represent numeric-type values. The data set surely does require a bit of manipulation, ahead of reviewing broad characteristics.

```
zomato = read_csv("zomato.csv")
str(zomato)
summary(zomato)
glimpse(zomato)
```

Before moving on, we should identify and determine how to manage erroneous and missing information in the data set. There may be entire columns that could be eliminated from further consideration. Features such as “url” and “phone” are not likely to contain useful information for our analysis of features that contribute to restaurant popularity. Other

variables provide unnecessary granular detail or duplicative information. For example, "address" is a more specific description of "location" or "listed_in(city)." The variable "reviews_list" seems redundant and more complex than its quantitative counterpart, "rate." Additionally, "menu_item" appears less informative than the "dish_liked" feature. The data set will be trimmed to exclude these problematic variables, and stored in a new object, zomato_reduced.

```
zomato_reduced = zomato %>% select(-url, -phone, -address, -reviews_list, -menu_item)
str(zomato_reduced)
summary(zomato_reduced)
glimpse(zomato_reduced)
```

The reduced data set is already looking much cleaner. We can now investigate null values.

```
colSums(is.na(zomato_reduced))
vim_plot = aggr(zomato_reduced, numbers = TRUE, prop = c(TRUE, FALSE),cex.axis=.7)
```

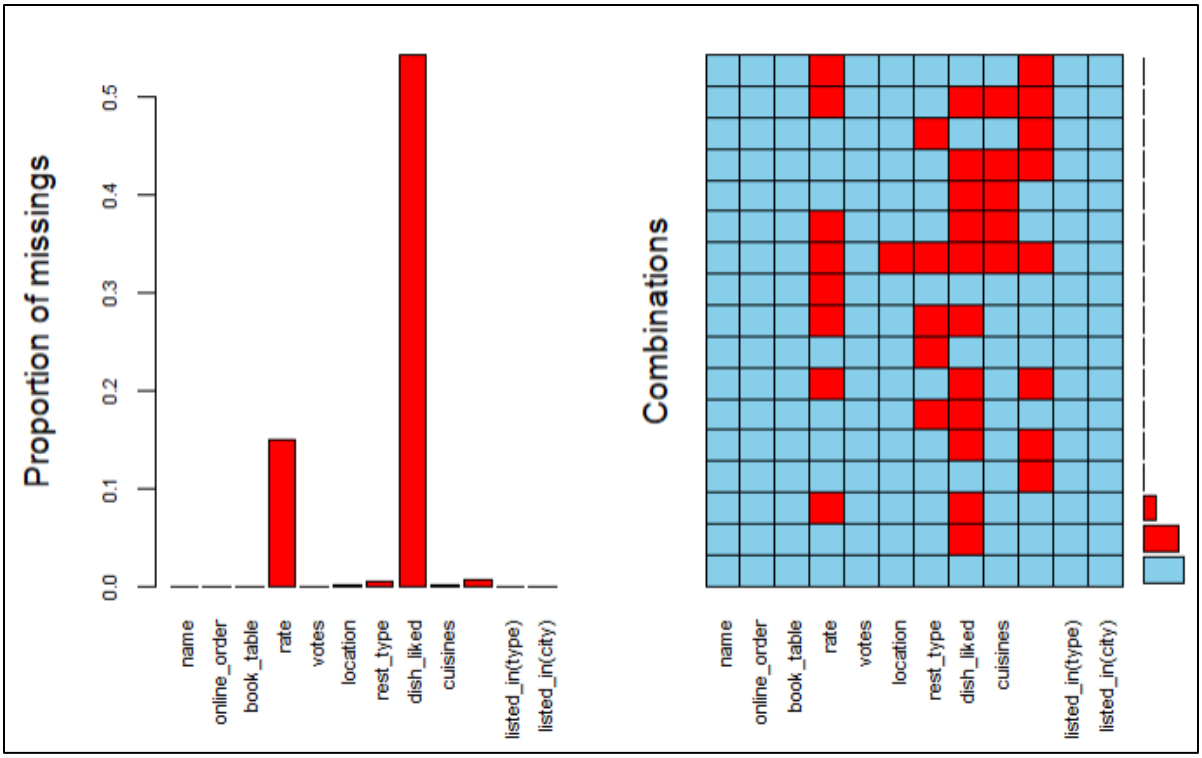


Figure C.1

Missing values are apparently present in six of the remaining twelve columns. Four columns ("location", "cuisines", "rest_type", and "approx_cost(for two people)") contain less than 1% missing information. The rows containing these null values can simply be eliminated. The other two columns "rate" and "dish_liked" have a higher prevalence of missing values (15% and 54.3%, respectively). Since "rate" is a response variable of interest, row-wise deletion of

null values seems appropriate. While “dish_liked” contains a large number of missing values, this variable is likely to contain information that is somewhat related to customer sentiment (expressed through “votes” and “rate”). For this reason, “dish_liked” will be set aside and individually analyzed a bit later to provide informative summary findings.

```
zomato_reduced = zomato_reduced %>% drop_na(rest_type, `approx_cost(for two
people)`,location, cuisines, rate)
```

Now that null values have been handled, it is time to take a closer look at unique values. With 41,263 observations in the reduced data set, duplicate values for restaurant name suggest that there are multiple reviews and/or multiple restaurant locations. Other variables are sure to have duplicate values as well. Identifying unique values can help determine the best method to explore each individual variable.

```
unique(zomato_reduced$rate)
```

Reviewing the unique values of “rate” shows that there are two values (“NEW” and “-“) that should be specifically excluded. This variable also contains a character grouping (“/5”) that can be disregarded, leaving only the base value of the rating.

```
zomato_reduced = zomato_reduced %>%
  filter(zomato_reduced$rate != "-")
zomato_reduced = zomato_reduced %>%
  filter(zomato_reduced$rate != "NEW")
zomato_reduced = zomato_reduced %>%
  mutate(rate = as.character(lapply(rate, sub, pattern = "/5", replacement = "")))%>%
  mutate(rate = as.numeric(rate))
```

The unique value counts of the remaining variables can be examined with this code:

```
count(as.data.frame(unique(zomato_reduced$name)))
count(as.data.frame(unique(zomato_reduced$online_order)))
count(as.data.frame(unique(zomato_reduced$book_table)))
zomato_reduced = zomato_reduced %>%
  mutate(online_order = as.factor(online_order)) %>%
  mutate(book_table = as.factor(book_table))
count(as.data.frame(unique(zomato_reduced$location)))
count(as.data.frame(unique(zomato_reduced$`listed_in(city)`)))
count(as.data.frame(unique(zomato_reduced$rest_type)))
count(as.data.frame(unique(zomato_reduced$`listed_in(type)`)))
count(as.data.frame(unique(zomato_reduced$dish_liked)))
count(as.data.frame(unique(zomato_reduced$cuisines)))
```

There are 6,602 unique restaurant names in the reduced data set. This supports the assumption of many restaurants having multiple locations and/or multiple reviews. The variables of “online_order” and “book_table” are binary and can easily be converted into factors. Juxtaposing “location” (92 unique values) and “listed_in(city)” (30 unique values) we

can see that the latter would be easier to categorically visualize. A similar comparison reveals that "listed_in(type)," with only 7 unique values should be easier to visualize than the 87 unique values in "rest_type." This assumes that these variable pairs can serve as surrogates for each other. The features "dish_liked" and "cuisines" have many unique values (5,239 and 2,487, respectively), making them both worthy of more in-depth review and potential parsing.

Visualization Methods

The visualization of "dish_liked" and "cuisines" will be accomplished using text mining and word clouds. The "dish_liked" column will be extracted and exported to Excel, where it can be saved locally and re-imported into R as a TXT file.

```
zomato_dishliked = zomato %>% dplyr::select(dish_liked) %>%
  drop_na(dish_liked)
write.csv(zomato_dishliked,
          file = 'zomato_dishliked.csv',
          row.names = FALSE,
          na = '-9999')
#(In Excel, save as .txt file and place in working directory)
text = readLines('zomato_dishliked.txt')
```

With this new TXT file saved in our working directory, the following block of code can be used to generate a bar plot and word cloud of the most frequent words found in the "dish_liked" column:

```
TextDoc = Corpus(VectorSource(text))
toSpace = content_transformer(function (x , pattern ) gsub(pattern, " ", x))
TextDoc = tm_map(TextDoc, toSpace, "/")
      TextDoc = tm_map(TextDoc, toSpace, "@")
TextDoc = tm_map(TextDoc, toSpace, "\\|")
TextDoc = tm_map(TextDoc, content_transformer(tolower))
TextDoc = tm_map(TextDoc, removeNumbers)
TextDoc = tm_map(TextDoc, removeWords, stopwords("english"))
TextDoc = tm_map(TextDoc, removePunctuation)
TextDoc = tm_map(TextDoc, stripWhitespace)
TextDoc = tm_map(TextDoc, stemDocument)
TextDoc_dtm = TermDocumentMatrix(TextDoc)
dtm_m = as.matrix(TextDoc_dtm)
dtm_v = sort(rowSums(dtm_m),decreasing=TRUE)
dtm_d = data.frame(word = names(dtm_v),freq=dtm_v)
barplot(dtm_d[1:5,]$freq, las = 2, names.arg = dtm_d[1:5,]$word,
        col ="deepskyblue3", main ="Top 5 most frequent words",
        ylab = "Word frequencies")
set.seed(123)
wordcloud(words = dtm_d$word, freq = dtm_d$freq, min.freq = 15,
          max.words=100, random.order=FALSE, rot.per=0.40,
          colors=brewer.pal(5, "Dark2"))
```

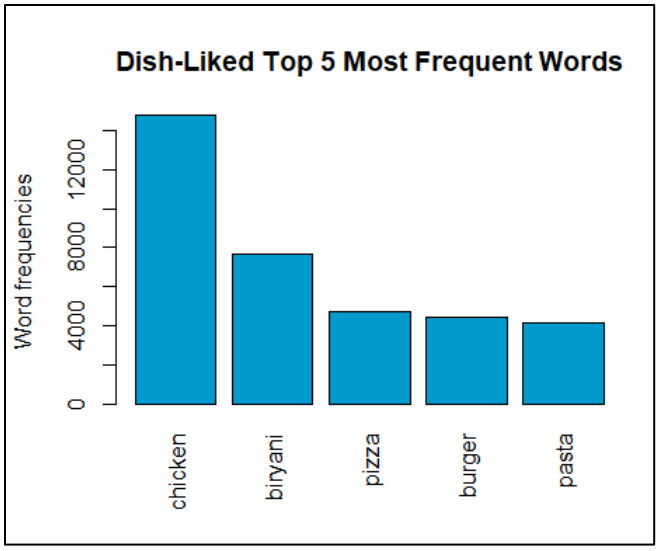


Figure C.2

Figure C.3

We can see that *chicken* leads the group in terms of word frequency within "dish_liked." The next most frequent word, *biryani*, appears about half as often. It is possible that chicken could be paired with biryani or the next most commonly occurring words (*pizza*, *burger*, and *pasta*). The word cloud also shows a handful of additional terms that are often mentioned (*sandwich, masala, paneer, salad, chocol[ate], coffee*, and *cocktail*). A similar analysis of "cuisines" can be performed with this nearly identical block of code:

```
zomato_cuisines = zomato_reduced %>%
  dplyr::select(cuisines)
write.csv(zomato_cuisines,
          file = 'zomato_cuisines.csv',
          row.names = FALSE,
          na = '-9999')
#(In Excel, save as .txt file in working directory)
text = readLines('zomato_cuisines.txt')
TextDoc = Corpus(VectorSource(text))
toSpace = content_transformer(function (x , pattern ) gsub(pattern, " ", x))
TextDoc = tm_map(TextDoc, toSpace, "/")
TextDoc = tm_map(TextDoc, toSpace, "@")
TextDoc = tm_map(TextDoc, toSpace, "\\|")
TextDoc = tm_map(TextDoc, content_transformer(tolower))
TextDoc = tm_map(TextDoc, removeNumbers)
TextDoc = tm_map(TextDoc, removeWords, stopwords("english"))
#Remove the word 'food' from consideration here
TextDoc = tm_map(TextDoc, removeWords, c("food"))
TextDoc = tm_map(TextDoc, removePunctuation)
TextDoc = tm_map(TextDoc, stripWhitespace)
TextDoc = tm_map(TextDoc, stemDocument)
TextDoc_dtm = TermDocumentMatrix(TextDoc)
dtm_m = as.matrix(TextDoc_dtm)
dtm_v = sort(rowSums(dtm_m),decreasing=TRUE)
dtm_d = data.frame(word = names(dtm_v),freq=dtm_v)
barplot(dtm_d[1:5,]$freq, las = 2, names.arg = dtm_d[1:5,]$word,
```

```
        col ="deepskyblue3", main ="Top 5 most frequent words",
        ylab = "Word frequencies")
set.seed(123)
wordcloud(words = dtm_d$word, freq = dtm_d$freq, min.freq = 20,
          max.words=100, random.order=FALSE, rot.per=0.40,
          colors=brewer.pal(5, "Dark2"))
```

It should come as no surprise that *indian* is the most frequent word in the "cuisines" variable from this data set. It is interesting, though, to note that *north* and *chines[e]* are the next most frequently appearing words, especially considering the culinary similarities between North Indian and Chinese food. Aside from the regional descriptors of cuisine, the next most commonly appearing word is *fast*, which seems to indicate that convenience and the speed of service or delivery may be important.

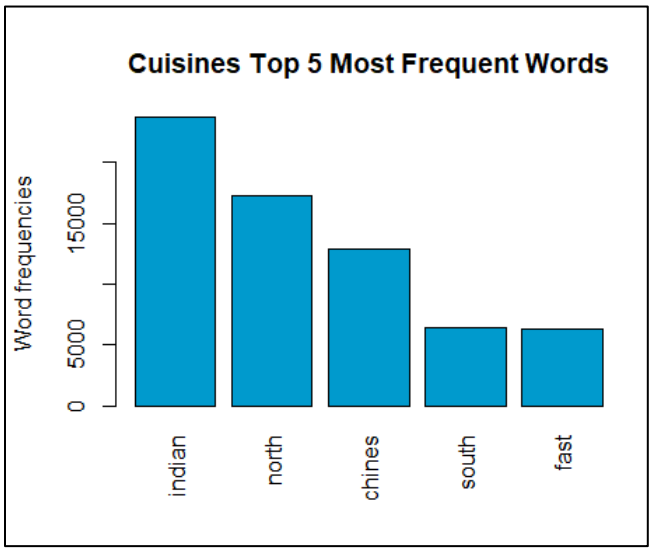


Figure C.4

Figure C.5

The remaining variables in the data set will be analyzed in Tableau:

```
#Export to CSV File for Tableau
write.csv(zomato_reduced,
          file = 'zomato_reduced.csv',
          row.names = FALSE,
          na = '-9999')
```

Rate:

The overall distribution of restaurant ratings is left-skewed, with the majority falling between 3.5 and 4.3 (out of 5). This makes sense, since low-rated restaurants would not likely remain in the market long-term. Seeing this spread provides a useful context for further review and targeted analysis.

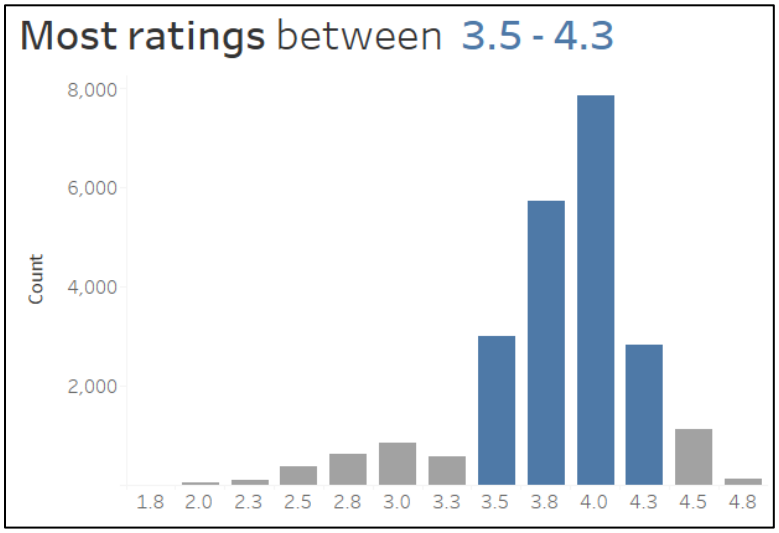


Figure C.6

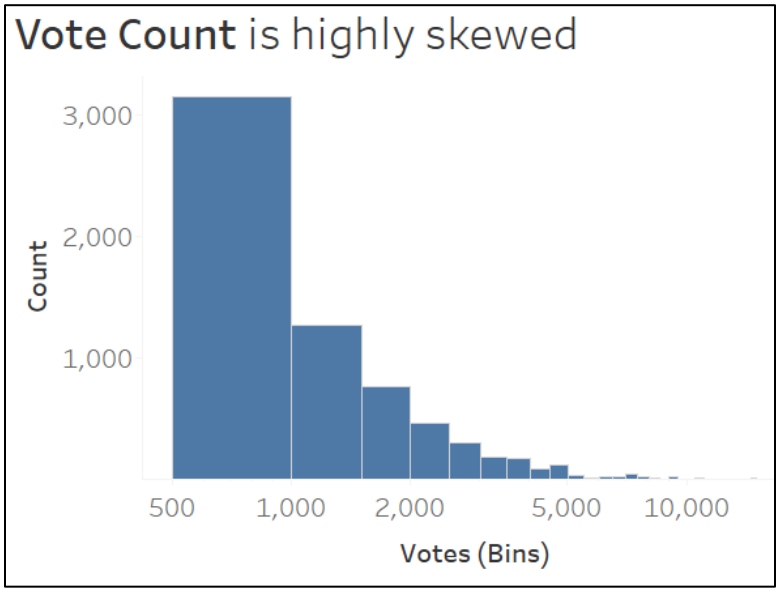


Figure C.7

Vote Count:

The highly skewed nature of this feature necessitated the application of a logarithmic scale. It is understandable that it would be difficult for a restaurant to amass a large number of votes. Without seeing the age of restaurants or any time-series trends of vote counts, it is difficult to draw many insights from this variable alone. However, paired with "rate," it could perhaps bolster (or hinder) a restaurant's overall favorability. Essentially, if a restaurant has received many votes and still maintains a high rating, that is a good sign.

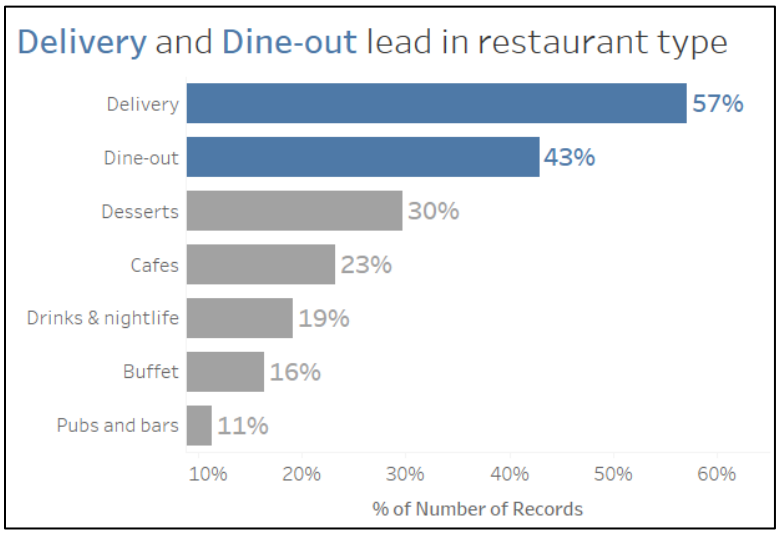


Figure C.8

Restaurant Type Distribution:

Examining the variable "Listed_in(type)," we can see that Delivery and Dine-out have a strong presence in the data set. These variables could be cross-referenced with others (such as "cuisines" or "dish_liked") to find other relationships.

Approximate Cost (for two people):

A histogram of this variable provides an opportunity to corral groups together by cost. Setting the bins to 600 provides the closest approximation of an even split. These three tiers can now be more closely examined, reviewing variable interaction within each cluster.

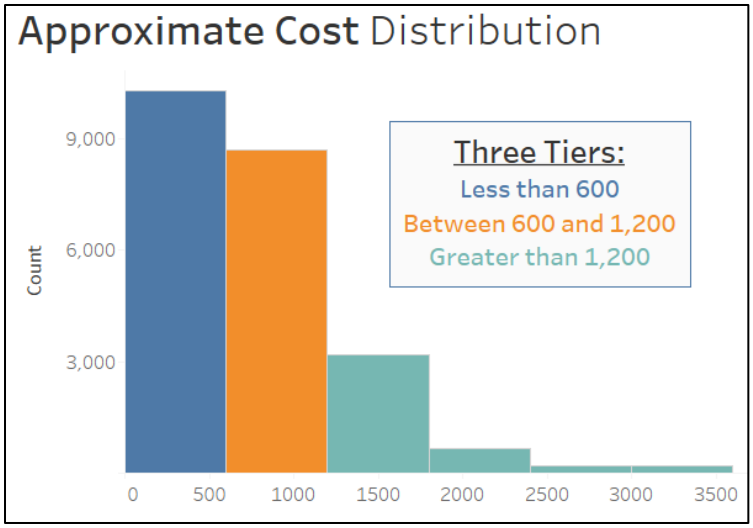


Figure C.9

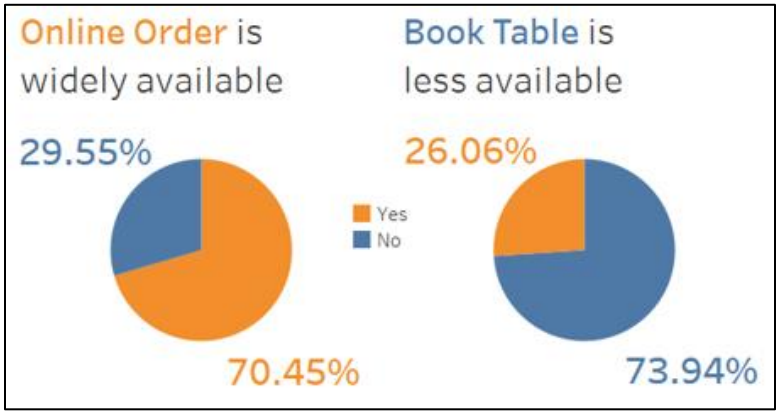


Figure C.10

Online Order and Book Table:

These variables are visualized together since they are both binary in nature. These pie charts show that, while ordering online is common in this data set, booking a table ahead of time is not. It will be interesting to see how these

variables interact with others in the data set. Some fluctuation with other variables is to be expected, but it is not yet clear if these two will be predictive of "votes" or "rate."

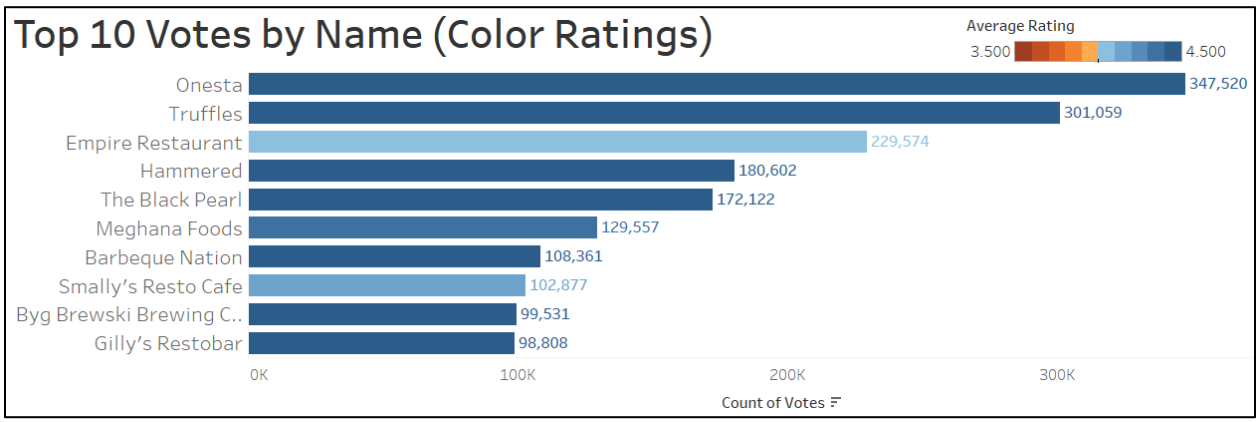


Figure C.11

Most Voted by Name: The visualization featured above involves multiple dimensions. First, the restaurant names with the Top 10 number of votes are listed in descending order. Onesta, Truffles, and Empire Restaurant hold a strong lead here. In addition to the vote count, the plot includes a color scheme to reflect the average rating of each restaurant. It is interesting to find some variation in ratings between the top restaurants. This shows that vote count may not necessarily be the primary driver of positive customer sentiment; ratings must also be taken into consideration.

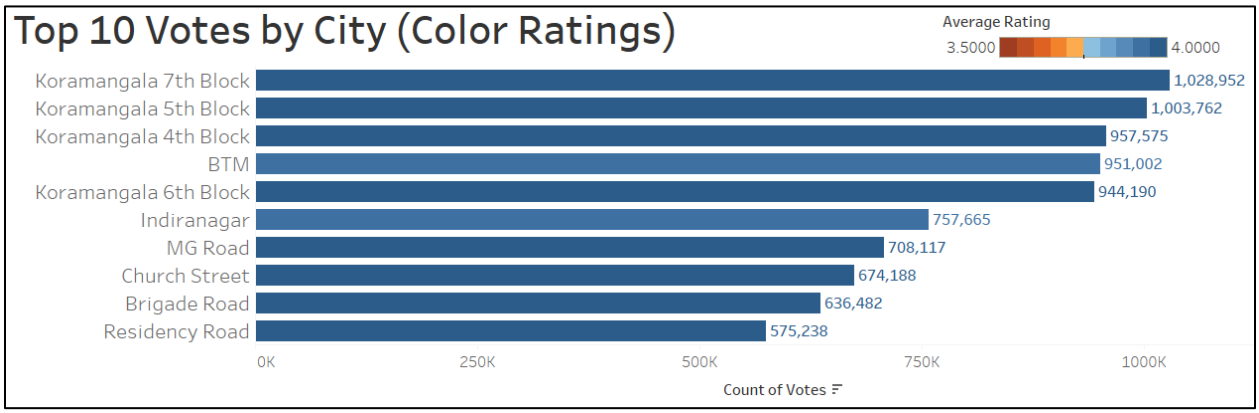


Figure C.12

Most Votes by City: This bar plot shows a similar view of the "Listed_in(city)" responses with the Top 10 number of votes, with each city's average rating represented by an orange-blue color scheme. Multiple blocks of Koramangala are featured prominently in this Top 10 grouping. In this case, there is not much difference in the average rating between city areas.

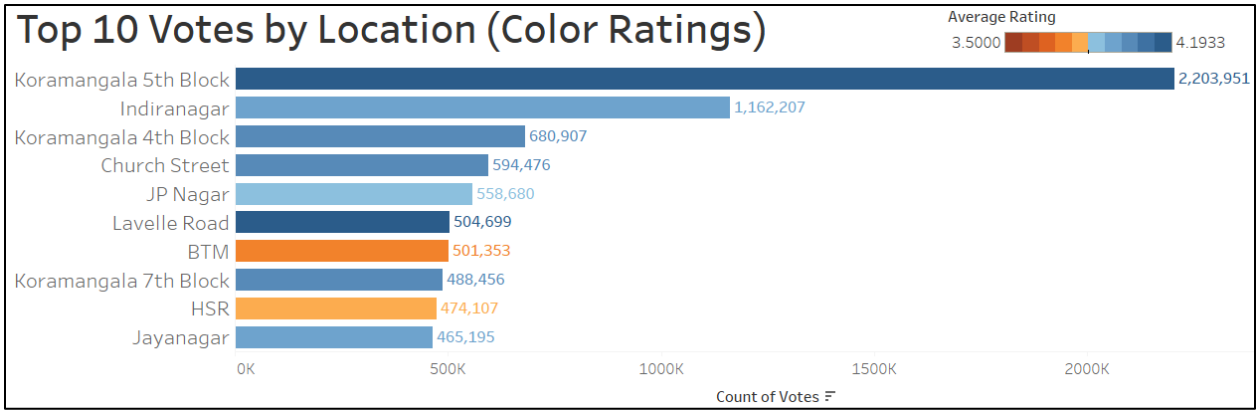


Figure C.13

Most Votes by Location: This visualization also applies a color scheme for ratings to the Top 10 locations by vote count. This bar plot shows a wider gap in ratings between some of the restaurants that have achieved a high vote count. Interestingly, the BTM location here appears to have a lower average rating than the BTM city area from the previous visualization. It will be important to take note of such apparent discrepancies, as we move into the next stage of analysis -- accounting for variable relationships and their interactions.

D. Cross Validation / Sampling

This stage of the analysis will include a more in-depth review of a stratified sampling of the data set. The aim is to investigate feature interaction within subsets of certain variables. This analysis will be conducted in Tableau, using a variety of data visualizations. This process will help inform the next phase of the analysis, in addition to evaluating whether the planned methods will be effective and if this revised data set is appropriate for the project scope.

The response variables of interest in this project, "rate" and "votes," are anticipated to reflect a similar view of customer satisfaction. The binary features of "book table" and "online order" are also expected to show an interconnected relationship. The approximate cost of restaurants will be studied with the availability of table bookings and online orders, as well as by restaurant type and city listing. Frequently occurring words in the "dish_liked" and "cuisines" variables will be analyzed, in order to determine any underlying patterns. Ratings distributions will also be examined by approximate cost, and top performing restaurants will be surveyed to find type listing trends.

Rate and Vote

First and foremost, it is helpful to investigate the relationship between rate and vote. Understanding how these features interact with each other will support and justify the selection of these as response variables.

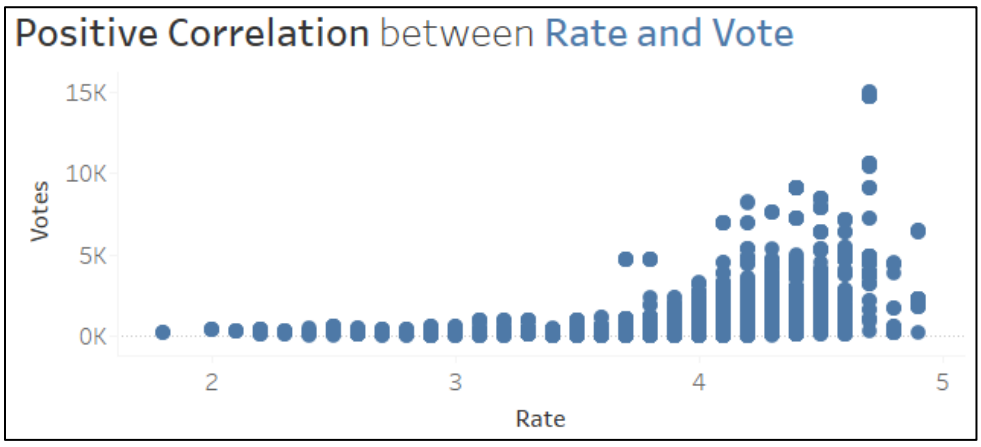


Figure D.1

The scatterplot above shows a positive, non-linear correlation between these two variables. This interaction is minimal in ratings below 3.5, but the interaction rises higher beyond that rating level. This connection seems intuitive, since restaurants with low ratings would not likely have as many votes (and many restaurants with a high occurrence of low

ratings could simply be run out of business). There is a substantial cluster of restaurants that have an average rating above 4.0 garnering a vote count below 5,000. There are a few exceptions, however, with outliers achieving a high rating while maintaining a vote count between 10,000 and 15,000.

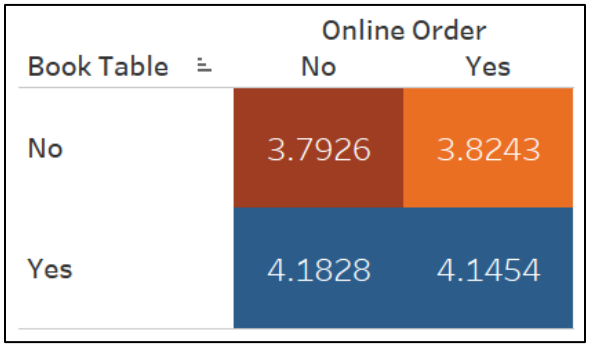


Figure D.2

Book Table and Online Order (Average Rate)

This matrix reveals that restaurants offering table reservations without online orders have a higher rating, 4.1828 on average. Table bookings seem to be more impactful to a restaurant's rating. Restaurants that offer neither reservations nor online orders are more likely to have a lower rating, 3.7926 on average. This naturally raises the question of which other variable(s) may interact with these convenience-related offerings.

Book Table and Online Order by Cost

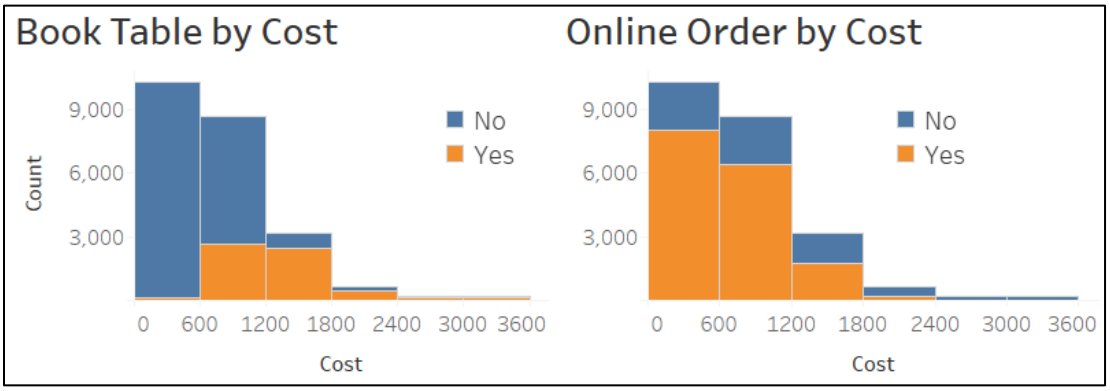


Figure D.3

This duo of stacked bar charts shows a somewhat unsurprising relationship between each of these variables and approximate restaurant cost. Low-cost restaurants are not likely to offer table bookings. They are, however, more likely to offer online ordering. Some mid-cost restaurants (about 30%) do offer reservations, but a more substantial number offer the option to order online. High-cost restaurants more uniformly offer table bookings, without the option to order online. Seeing how the options of table bookings and online ordering interact with each other and approximate cost will inform how a new restaurant decides to structure its own options within each cost tier.

Cost by Type Listing

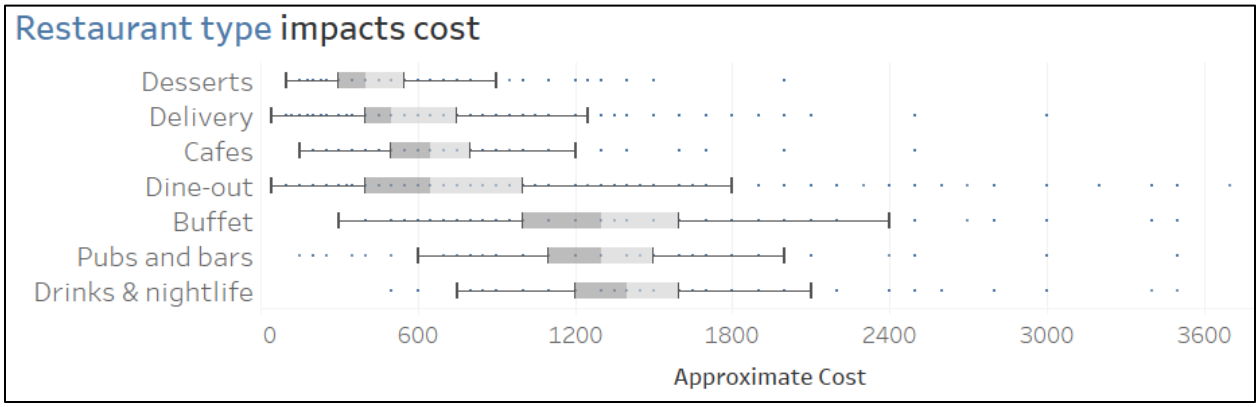


Figure D.3

The visualization above shows a clear distinction between the cost distributions of restaurant types. Restaurants that focus on desserts, delivery, and lighter cafe fare tend to have a lower cost. This lower-cost group of restaurant types also has a narrower range than the next series of types. Restaurants in the dine-out, buffet, pubs and bars, and drinks & nightlife types generally have a higher cost, distributed over a wider range.

Cost by City Listing

The data set contains thirty city listings. To best visualize the cost distribution of these, it is helpful to review the characteristics of smaller subsets by area cost. Knowing which areas have higher or lower overall costs should help a new restaurant avoid running the risk of offering an unsuitable cost structure.

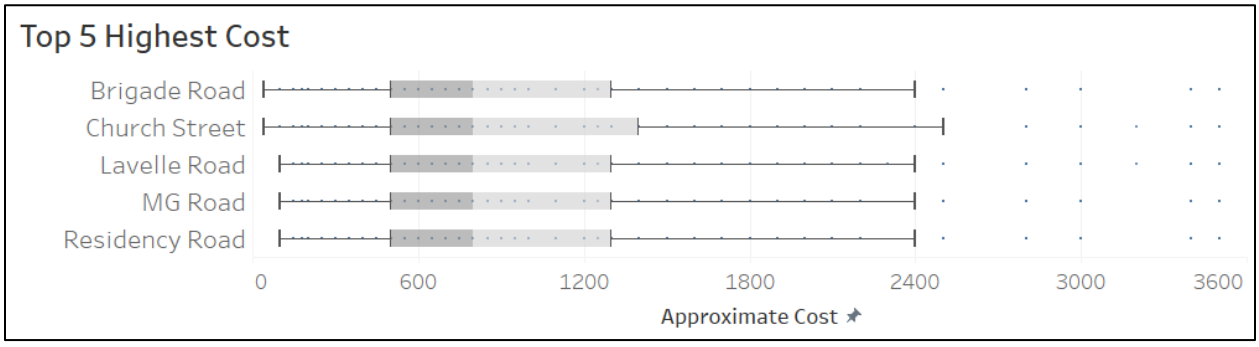


Figure D.4

The highest-cost city listings have cost distributions that are very similar to each other. Their lowest cost value is very low, which can be expected with any location. However, their interquartile range stretches from just below 600 to just above 1200. There are also a handful of outliers in all of these city listings that cost as much as 3600 for a party of two.

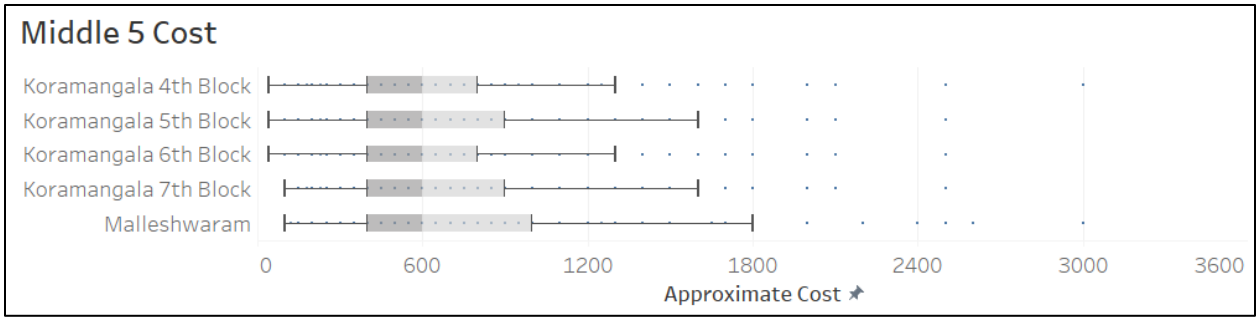


Figure D.5

This group of city listings from the middle of the comparison shows a slightly lower cost. The median here is 600. There are very low costs included in this set as well. However, the outliers on the higher end of the range do not rise above 3000. It is interesting to note that four blocks of Koramangala are represented right here in this central cost grouping.

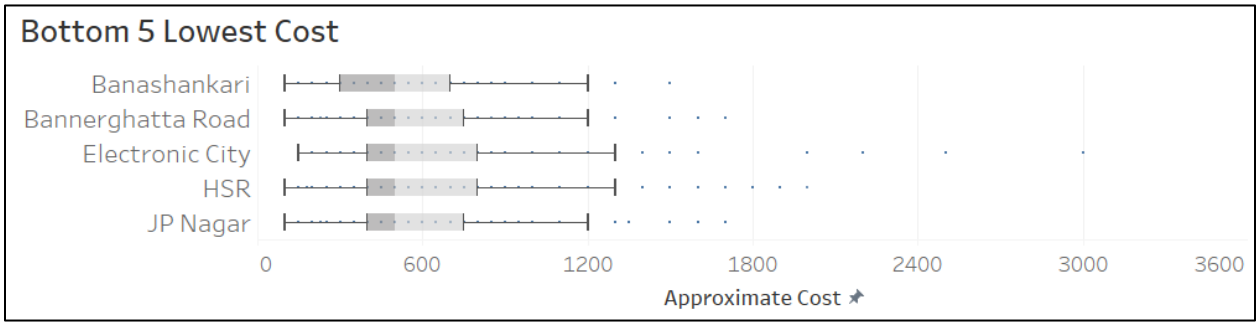


Figure D.6

The five lowest-cost city listings have a median cost below 600, with the interquartile range mostly stretching from 400 to 800. There is one area, Electronic City, with a couple of surprising outlier costs above 2400. However, for the most part, these city listings predominantly offer lower cost restaurants.

Ratings by Top Dishes-Liked

The word cloud analysis conducted previously showed these most commonly occurring terms in the "dish-liked" variable: *chicken*, *biryani*, *pizza*, *burger*, and *pasta*. Examining the ratings distribution of restaurants that feature these preferred dishes provides a layer of additional context.

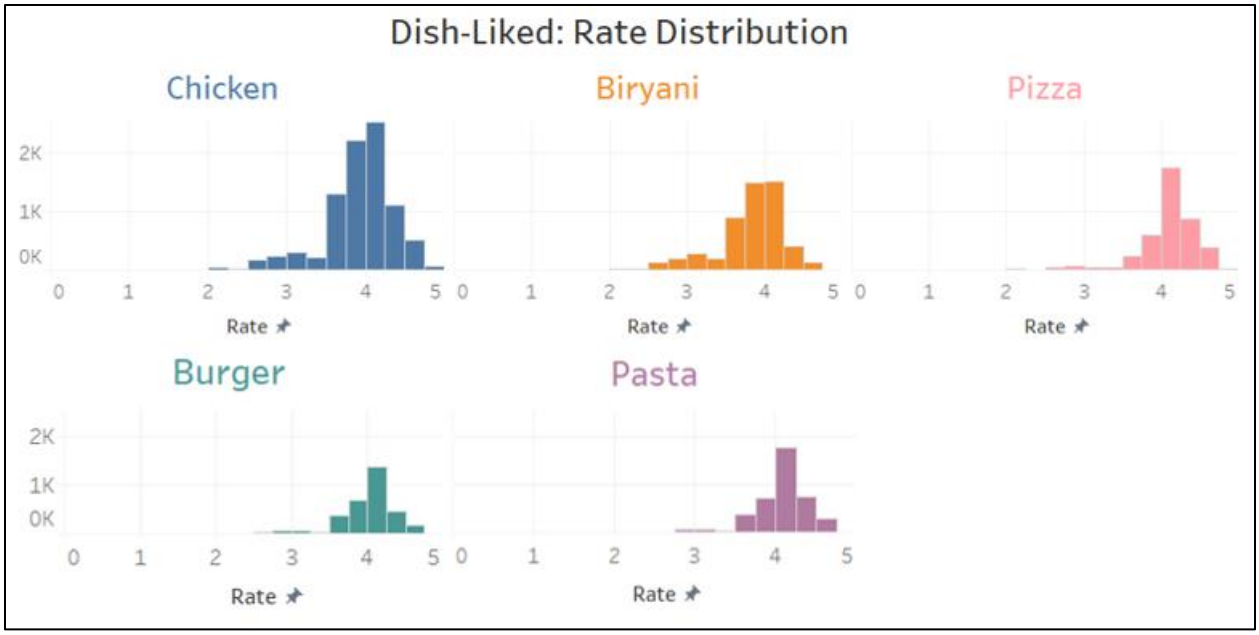


Figure D.7

The distributions of these dishes-liked show the majority appearing alongside ratings that hover around 4.0. *Chicken* and *biryani* have a handful of mentions in ratings between 2.5 and 3.5. Aside from that, though, most dish-liked mentions occur with higher ratings. It is possible that restaurant reviewers are less likely to indicate any dish-liked at all when leaving a fair or mediocre rating. *Pizza* appears to have a slightly left-skewed distribution, with most of its mentions aligning with ratings above 4.0. *Burger* shows a somewhat opposite trend, with the majority of its mentions accompanying ratings between 3.5 and 4.0.

Ratings by Top Cuisine

A similar word cloud analysis of the "cuisines" variable showed these types of food being offered by restaurants in the data set: North Indian, Chinese, South Indian, and Fast. Examining these most-featured cuisines by rate should provide additional insight.

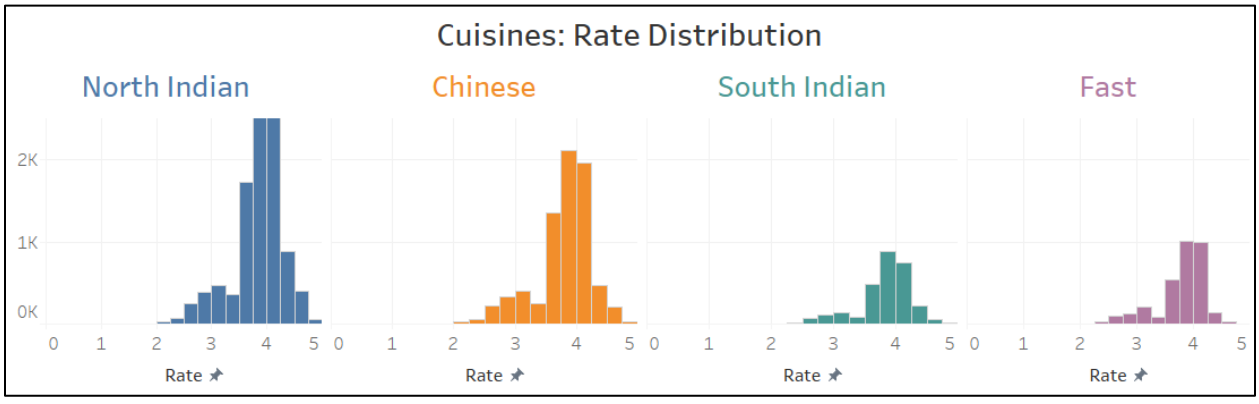


Figure D.8

Each of these cuisines has a similar rate distribution. North Indian and Chinese both have a small crest of votes encircling 3.0, with a much larger swell around 4.0. South Indian and Fast show a similar trend, but on a much smaller scale. It is striking to see the difference between the number of mentions between these cuisines reiterated here.

Ratings by Cost Tier

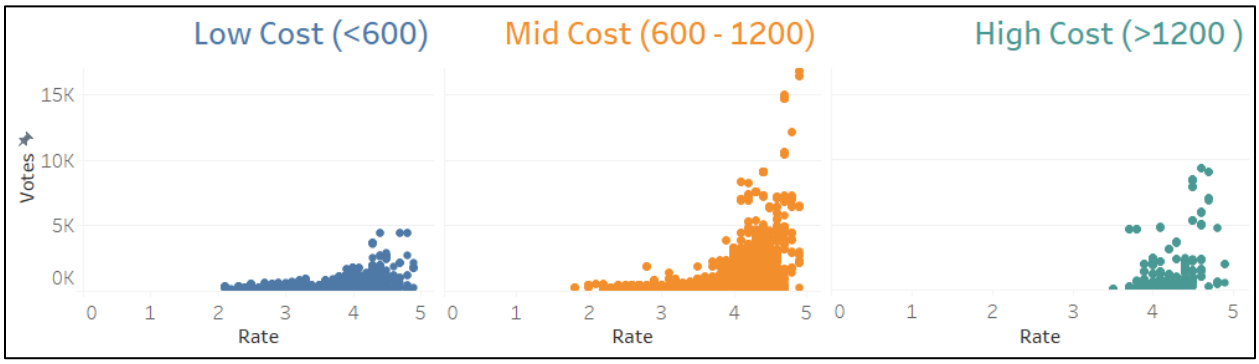


Figure D.9

Each cost tier has its own rate-to-vote characteristics. The low-cost segment holds fewer than 2,500 votes for most of its restaurant ratings. For ratings above 4.0 in this segment, there are a few that nearly reach 5,000 votes, but none exceed that figure.

The mid-cost segment includes some ratings below 2.0, which is worse than any of the restaurants in the low-cost group. Otherwise, these two segments perform similarly with votes and ratings up to 3.5. However, the mid-cost segment has quite a few instances of restaurants rated above 4.0 that also exceed five-thousand, ten-thousand, or even fifteen-thousand votes. The mid-cost group also has a higher concentration of restaurants scoring a rating above 4.5.

The high-cost segment can boast of having no ratings that fall below 3.5. Most ratings in this group land between 4.0 and 4.8. It is noteworthy that most ratings of 4.5 to 4.8 in this segment also manage to secure between 5,000 and 10,000 votes.

Type Listings of Top-Performing Restaurants

Highest-rated and top-voted restaurants were identified in the low-cost, mid-cost, and high-cost tiers. Taking a glimpse of the restaurant type listing of these top performers might uncover what makes them so successful. The following visualizations show a percentage distribution of the type listings associated with each restaurant:

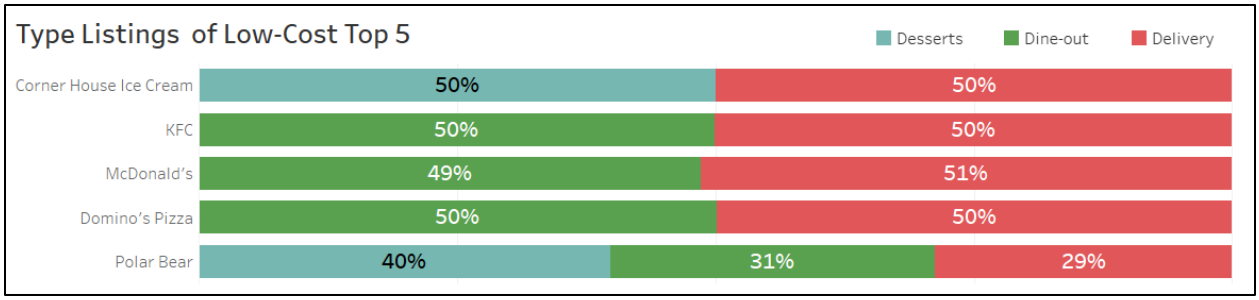


Figure D.10

There is little variation in the type listing of the low-cost segment's top performers. Dine-out and Delivery hold a majority stake in this space, leaving only some room for Desserts. Judging by the restaurant names here, we can also infer popular menu items (ice cream, chicken, burgers, and pizza).

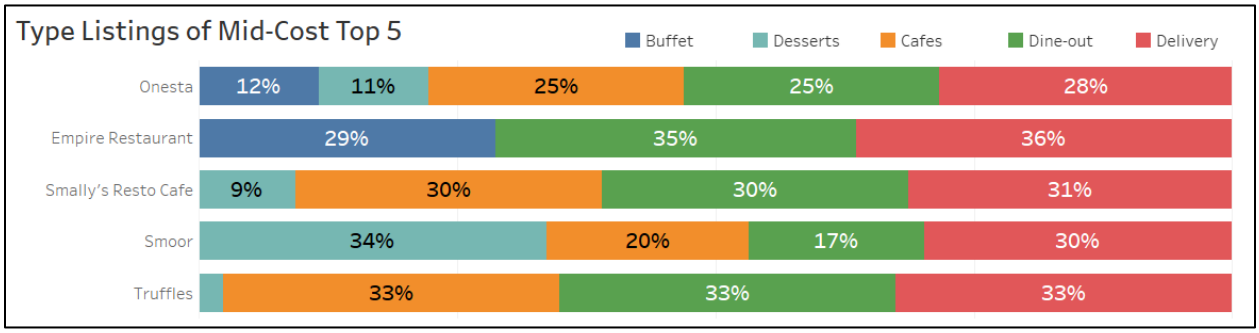


Figure D.11

Within the mid-cost segment, there is more variety in terms of restaurant type listings. For example, the best performing restaurant in this segment (Onesta) is listed in Buffet, Desserts, Cafes, Dine-out, and Delivery. The next best-performer, Empire Restaurant, is listed in Buffet, which signals a variety of cuisines being offered. It is interesting to find restaurants in this segment being either listed in a number of different types or offering a wide selection of foods.

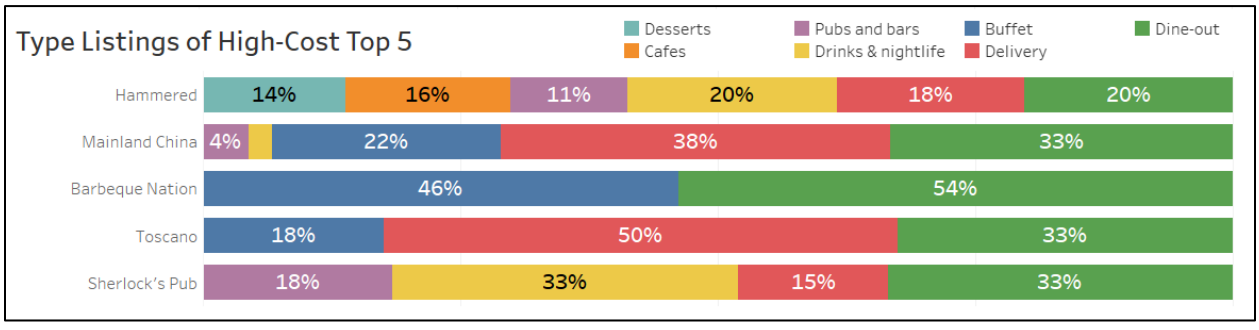


Figure D.12

Restaurants in the high-cost segment largely incorporate multiple type listings or cuisine types as well. The best-performing restaurant in the high-cost segment (Hammered) exhibits this versatility, having been listed in six different type categories. Three of the other top performers in this high-cost segment are listed in Buffet, highlighting the popularity of culinary variety in this group.

Final Notes on Data Preparation

We have cleaned the data set and analyzed how features interact with each other. This exercise in data preparation has held-out certain variables from the data set, due in part to the underlying assumption that some features were duplicative or overly verbose. After reviewing a cross-sectional analysis of the reduced data set, we can confirm that the variables remaining will adequately inform the next stage. This will involve using predictive analytics models to determine the degree to which features contribute to the response variables of “rate” and “votes.”

The data preparation from this stage will also provide the majority of information required for completion of the final task, in which a prescriptive analytics solution will offer recommendations for a new restaurant entering the Bengaluru market. However, some additional context may be required to define constraint parameters that cannot easily be extrapolated from the Zomato data set. This information will be gathered and reported in the final stage of the analysis.

E. Predictive Model Building

The response variables of “rate” and “vote” reflect a similar view of customer sentiment in the Zomato data set. If we can determine the predictors of ratings and vote counts, this will help with making recommendations for a new restaurant entering the Bengaluru market.

In order to set the stage for predictive modeling, the data set requires some slight adjustments. We have already removed null values and eliminated non-essential variables. However, to take advantage of information gleaned from the word-cloud analysis of the “dish_liked” and “cuisines” variables, a series of new columns will be added to the data set. These new columns will employ one-hot encoding to create binary values that indicate whether(1) or not(0) a most-frequently occurring word exists in that row. The dishes liked will include the following: *chicken*, *biryani*, *burger*, *pizza*, *pasta, sandwich, chocol[ate], masala, and fri[es]*. The cuisines will include *North Indian*, *South Indian*, *Chines[e]*, *Fast*, *Continental*, and *Cafe*. This has been performed using the following code in R:

```
zomato_reduced$DL_Chicken = ifelse(grepl("Chicken",zomato_reduced$dish_liked),1, 0)
zomato_reduced$DL_Biryani = ifelse(grepl("Biryani",zomato_reduced$dish_liked),1, 0)
zomato_reduced$DL_Pizza   = ifelse(grepl("Pizza",zomato_reduced$dish_liked),1, 0)
zomato_reduced$DL_Burger  = ifelse(grepl("Burger",zomato_reduced$dish_liked),1, 0)
zomato_reduced$DL_Pasta   = ifelse(grepl("Pasta",zomato_reduced$dish_liked),1, 0)
zomato_reduced$DL_Sand    = ifelse(grepl("Sandwich",zomato_reduced$dish_liked),1, 0)
zomato_reduced$DL_Choc    = ifelse(grepl("Chocol",zomato_reduced$dish_liked),1, 0)
zomato_reduced$DL_Masala  = ifelse(grepl("Masala",zomato_reduced$dish_liked),1, 0)
zomato_reduced$DL_Fri     = ifelse(grepl("Fri",zomato_reduced$dish_liked),1, 0)
zomato_reduced$Cuis_NI    = ifelse(grepl("North Indian",zomato_reduced$cuisines),1, 0)
zomato_reduced$Cuis_SI    = ifelse(grepl("South Indian",zomato_reduced$cuisines),1, 0)
zomato_reduced$Cuis_CH    = ifelse(grepl("Chines",zomato_reduced$cuisines),1, 0)
zomato_reduced$Cuis_FA    = ifelse(grepl("Fast",zomato_reduced$cuisines),1, 0)
zomato_reduced$Cuis_Cont  = ifelse(grepl("Continent",zomato_reduced$cuisines),1, 0)
zomato_reduced$Cuis_Caf   = ifelse(grepl("Cafe",zomato_reduced$cuisines),1, 0)
```

The variables remaining in the finalized data set will include this list of binary variables along with *approx_cost(for two people)*, *listed_in(city)*, *listed_in(type)*, *book_table*, and *online_order*.

Prediction Methods

The predictive analysis in this report has been performed in JMP Pro 16 using a wide array of methods, listed below:

Ordinary Least Squares	Random Forest
Stepwise Forward Regression	Boosted Tree
Stepwise Backward Regression	Neural Nets, with various configurations
Adaptive Lasso Regression	- One Layer, One Node (1-0-0), 40 Models, 20 Tours
Adaptive Elastic Net Regression	- One Layer, One Node (3-0-0), 40 Models, 20 Tours
Standard Decision Tree	- One Layer, Three Nodes (3-3-3) 40 Models, 20 Tours

Ordinary Least Squares: This is often used as a benchmark for analysis. It is not expected to outperform the other models in the group. This model attempts to minimize the differences (squared) of the predicted and actual values. By retaining all of the data set variables in its results, this method tends to overfit models to the training set.

Stepwise Forward and Backward: These alternative approaches are helpful in determining dependent variables. They each take incremental approaches to improving the final model. One concern with these methods is multicollinearity, since individually removing/adding variables might impact the inclusion/exclusion of other variables further down the chain.

Adaptive Lasso: Using the absolute value of a determined penalty, this method shrinks the variables that contain little information, so that those are not included in the final model. This method also tends to eliminate redundant variables that are highly correlated with others. Being adaptive, it takes into consideration the results of Ordinary Least Squares Regression before applying its penalization factor to unimportant variables.

Adaptive Elastic Net: This also considers the results of Ordinary Least Squares Regression before applying its penalized regression approach. However, this method applies both an absolute (Lasso Regression) and squared (Ridge Regression) penalization value to uninformative variables. While unimportant variables are eliminated in the resulting model, highly correlated variables are normally retained.

Standard Decision Tree: This method creates a classification tree by separating the variables at each node, according to the largest disparity. This greedy approach does not always provide an optimal solution.

Random Forest: This method creates many decision trees, using a subset of variables to enforce an element of randomness. The resulting array of uncorrelated decision trees make predictions, and the one that makes the most accurate predictions is the model that is chosen.

Boosted Tree: This method applies a boosting approach to a decision tree framework. This incremental approach collects the residuals, or errors, as each model is estimated. Those errors are then factored into the estimation of subsequent model estimates, after which an average of all models is used.

Neural Nets: These methods are helpful for modeling very complex non-linear relationships. Designed to simulate how the human brain processes information, neural net models send inputs through nodes, where a transformation function is applied. The assorted configurations differ in complexity, varying the number of nodes and selected activation functions (Hyperbolic Tangent, Linear, and Gaussian).

Procedure

Since the Zomato data set in this analysis is cross-sectional in nature (as opposed to being time series), the cross-validation approach for this analysis incorporates randomness. The data set has been split into three parts. The *Make Validation Column* facility in JMP Pro 16 was selected, applying a random seed of 123. Implementing a 60-20-20 split of the 41,263 total rows, 24,758 rows were used to train the models, 8,253 rows were tied to validation, and 8,252 rows were set aside for testing.

The “rate” and “votes” features were individually selected in JMP Pro 16 as the Y variables. All of the other variables in the data set were included as candidate predictors. The Validation Column was implemented, and the analysis was conducted using all methods. The random seed of 123 was used where applicable (e.g., configuring Neural Net and Boosted Tree models). As predictions from each model were generated, they were saved and stored in new columns in the data set. The figures below show a comparison of how all models performed with the test set.

Model Comparisons

Measures of Fit for "rate"

Creator	RSquare	RASE	AAE
Ordinary Least Squares	0.3613	0.3518	0.2701
Stepwise Forward	0.3613	0.3518	0.2702
Stepwise Backward	0.3613	0.3518	0.2701
Adaptive Lasso	0.3613	0.3518	0.2701
Adaptive Elastic Net	0.3613	0.3518	0.2701
Standard Decision Tree	0.4147	0.3368	0.2517
Random Forest	***0.4511***	***0.3262***	***0.2463***
Boosted Tree	***0.4869***	***0.3153***	***0.2336***
Neural Net 1-0-0	0.3944	0.3426	0.2605
Neural Net 3-0-0	0.4470	0.3274	0.2478
Neural Net 1-1-1	0.4331	0.3315	0.2506

Measures of Fit for "votes"

Creator	RSquare	RASE	AAE
Ordinary Least Squares	0.2576	754.09	321.51
Stepwise Forward	0.2600	752.85	318.36
Stepwise Backward	0.2576	754.09	321.51
Adaptive Lasso	0.2576	754.09	321.51
Adaptive Elastic Net	0.2576	754.09	321.51
Standard Decision Tree	0.5371	595.45	236.94
Random Forest	0.4875	626.51	242.11
Boosted Tree	***0.7253***	***458.76***	***194.39***
Neural Net 1-0-0	0.3477	706.82	280.70
Neural Net 3-0-0	***0.7027***	***477.16***	***227.30***
Neural Net 1-1-1	0.5892	560.95	247.47

Figure E.1

The ideal models in these comparisons should have the highest RSquare, the lowest RASE(Root Average Square Error), and the lowest AAE(Average Absolute Error) values. In the comparison for "rate," the Boosted Tree and Random Forest models were the top performers, with RSquare values of 0.4869 and 0.4511 respectively. These models also had the lowest RASE values (0.3153 for Boosted Tree, 0.3262 for Random Forest), and the lowest AAE values (0.2336 for Boosted Tree, 0.2463 for Random Forest).

The top performing models in the comparison for "votes" turned out to be the Boosted Tree and the single-layer Neural Net model with 3 nodes (using a hyperbolic tangent activation function). The RSquare value of these models (0.7253 for Boosted Tree, 0.7027 for Neural Net) was substantially higher than the next highest (0.5892 from another Neural Net model configuration). These RSquare values were also much better than those of the model comparison for "rate," indicating a greater predictive capability of "votes."

It is interesting to find that, especially with the model comparison for "rate," less complex models are able to compete with more complex ones. The Random Forest and Boosted Tree models took mere seconds to run, while the Neural Net models took many hours. The top-performing models for both "rate" and "votes" will be reviewed in further detail.

Model Interpretation - Rate

Reviewing the parameters of the winning models for "rate," the Boosted Tree model contained 200 layers, with 19 splits per tree. The Random Forest model involved 100 trees, with 14 terms sampled per split.

Variable Importance - Rate

Variable importance analysis reveals some divergence in the predicted results between the winning models. One trait these models share is a large disparity between the main effects and total effects for their important variables. This indicates that there is a substantial amount of interaction between variable effects, and that not just one feature is important. This may be one reason for the different ordering structures of the models, shown below.

Boosted Tree

Column	Main Effect	Total Effect
approx_cost(for two people)	0.097	0.240
listed_in(city)	0.032	0.221
Cuis_SI	0.032	0.186
Cuis_FA	0.021	0.137
DL_Pasta	0.035	0.132
DL_Sand	0.030	0.132
DL_Chicken	0.019	0.126
Cuis_Cont	0.024	0.117
Cuis_Caf	0.022	0.108
DL_Masala	0.014	0.103
book_table	0.021	0.100
Cuis_NI	0.016	0.095
DL_Choc	0.016	0.088
DL_Biryani	0.026	0.077
DL_Pizza	0.013	0.072
online_order	0.016	0.065
DL_Burger	0.015	0.064
Cuis_CH	0.014	0.061
DL_Fri	0.012	0.042
listed_in(type)	0.006	0.014

Random Forest

Column	Main Effect	Total Effect
book_table	0.410	0.513
approx_cost(for two people)	0.243	0.332
listed_in(city)	0.031	0.077
Cuis_CH	0.032	0.071
online_order	0.028	0.069
DL_Choc	0.027	0.060
DL_Pizza	0.017	0.042
DL_Chicken	0.019	0.042
Cuis_NI	0.013	0.024
DL_Pasta	0.009	0.023
Cuis_SI	0.009	0.019
DL_Sand	0.007	0.018
Cuis_Cont	0.006	0.011
Cuis_Caf	0.004	0.010
DL_Burger	0.004	0.007
Cuis_FA	0.002	0.004
DL_Masala	0.002	0.004
DL_Biryani	0.001	0.003
DL_Fri	0.001	0.002
listed_in(type)	0.001	0.002

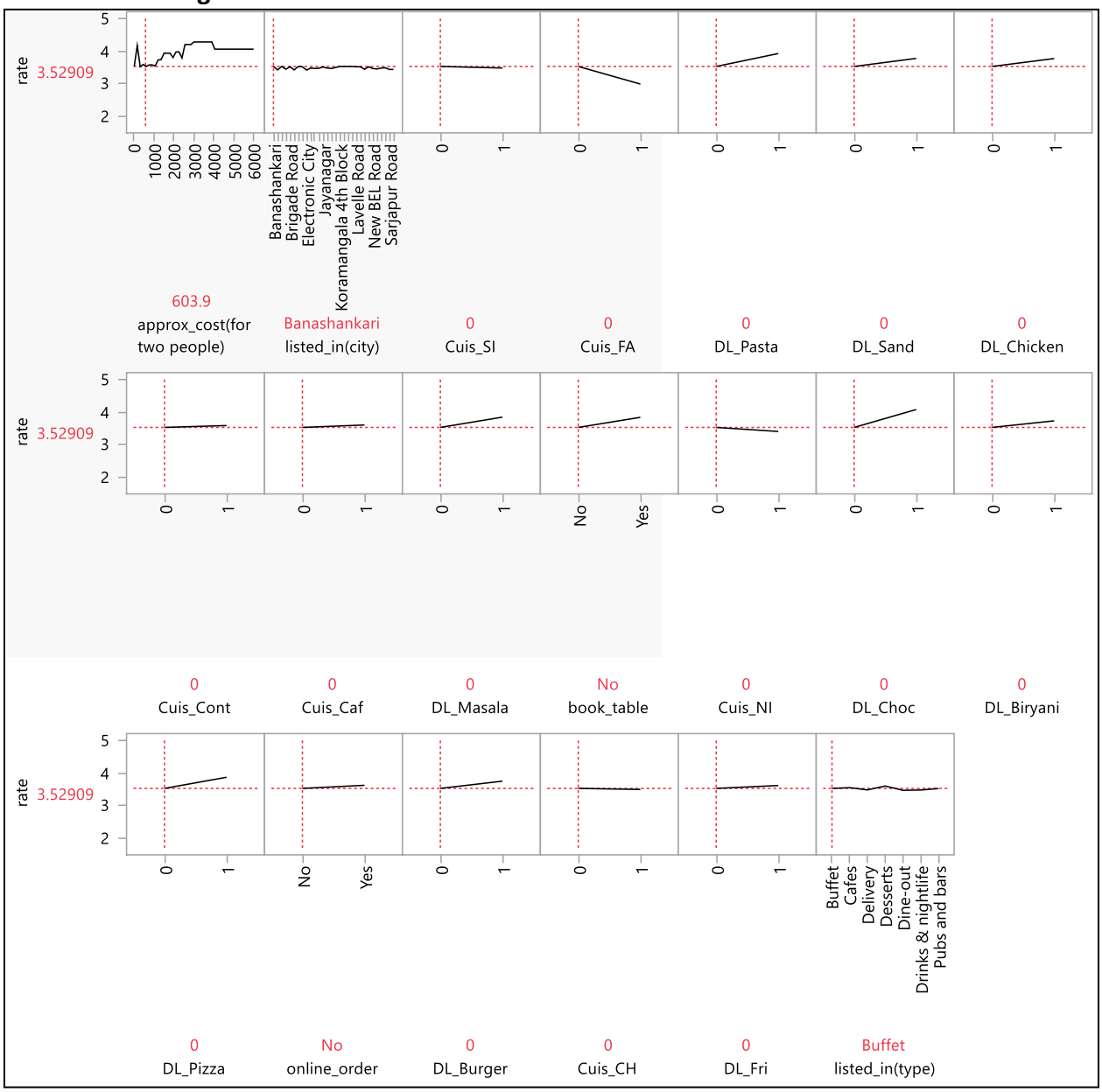
Figure E.2

The Boosted Tree and Random Forest models both include “approx_cost(for two people)” in the most important features, with total effects of 24% and 33.2% respectively. However, the Random Forest model places a stronger emphasis on “book_table,” with its dominant total effect of 51.3%. The Boosted Tree model has assigned “book_table” a total effect of only 10%. The next important variable in both models is “listed_in(city),” with the Boosted Tree model giving this variable a total effect of 22.1% (only 7.7% in the Random Forest model).

Prediction Profilers - Rate

Interactions between variables, as well as their effects on “rate” can be examined through model profilers. Each of these profilers will be reviewed individually.

Boosted Tree - Figure E.3

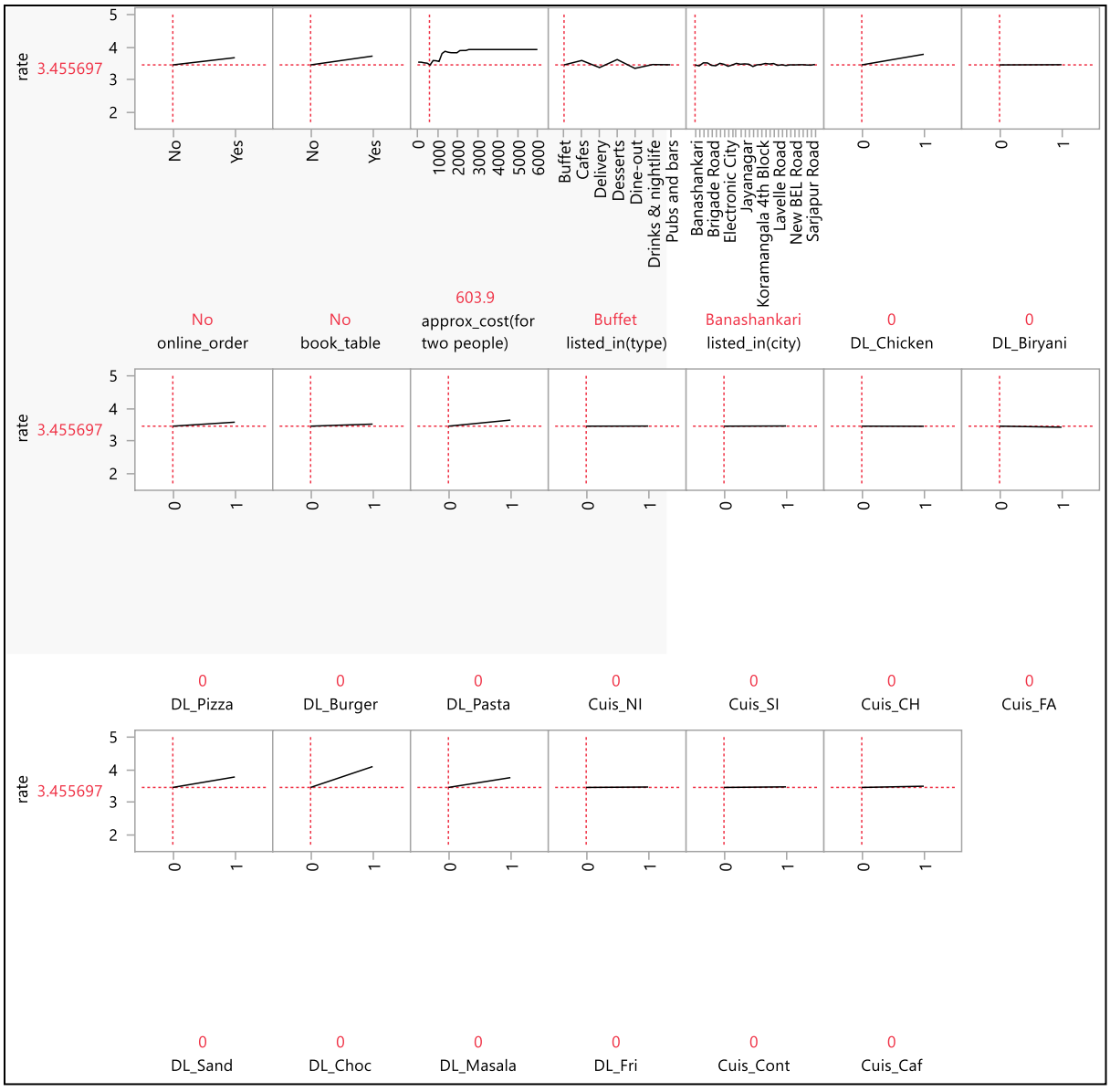


The Boosted Tree profiler helps visualize how certain features impact the “rate” response variable. For example, the steep slopes in variables such as *DL_Choc*(chocolate dish-liked) and *DL_Pizza*(pizza dish-liked) show how these liked dishes are significantly important to a higher rating (assuming, among other things, an approximate cost of 603.9). Flipping through approximate cost levels demonstrates how corresponding features within each range seem to

impact a restaurant's rating. The same can be said for toggling between the availability of commonly liked dishes, table bookings, online ordering, etc.

The profiler confirms the idea of generally rising rate values with an increase to approximate cost. This visual representation of feature interaction also illustrates how few variables are solely predictive of "rate." This aligns with the disparity between main and total effects presented in the variable importance analysis. Only with the appropriate adjustment of other variables do features like "listed_in(city)" rise to prominence.

Random Forest - Figure E.4



The Random Forest profiler shows a fair amount of variability within the average approximate cost of 603.9. Restaurants with dishes-liked that include *chocolate*, *burger*,

sandwich, *chicken*, and *pasta* tend to have a higher rating. At this level of approximate cost, there are also marginally higher ratings for the *Cafes* and *Desserts* restaurant type listings. The availability of online orders and table bookings also increase "rate."

Adjusting the levels of variables in this profiler makes it apparent that not all of these relationships are consistent. At different at cost levels, for example, the availability of online orders and table bookings can have a negative correlation with ratings. This volatility in feature interaction mostly occurs when adjusting values of three variables: "approx_cost(for two people)", "listed_in_(type)", and "listed_in(city)."

Model Interpretation - Votes

The top-performing model for predicting "votes" also utilized a Boosted Tree method containing 200 layers, with 19 splits per tree. The next best model involved a Neural Net that applied a zero-centered, S-shaped hyperbolic tangent activation function to three nodes. This hyperparameter tuning allowed for a non-linear relationship with outliers while approximating a linear regression for mid-range values.

Variable Importance - Votes

Boosted Tree

Column	Main Effect	Total Effect
approx_cost(for two people)	0.070	0.364
DL_Choc	0.091	0.220
book_table	0.046	0.207
listed_in(city)	0.023	0.172
Cuis_Cont	0.029	0.126
DL_Chicken	0.020	0.125
online_order	0.028	0.103
Cuis_CH	0.028	0.096
DL_Pizza	0.015	0.081
DL_Burger	0.017	0.077
DL_Pasta	0.027	0.075
Cuis_Caf	0.028	0.067
DL_Sand	0.025	0.062
listed_in(type)	0.013	0.039
DL_Masala	0.011	0.039
Cuis_SI	0.010	0.034
DL_Fri	0.013	0.029
DL_Biryani	0.006	0.016
Cuis_FA	0.006	0.014
Cuis_NI	0.005	0.014

Neural Net

Column	Main Effect	Total Effect
approx_cost(for two people)	0.169	0.743
listed_in(city)	0.030	0.429
DL_Choc	0.093	0.352
DL_Sand	0.039	0.308
DL_Pizza	0.018	0.264
Cuis_Cont	0.015	0.258
DL_Pasta	0.021	0.256
DL_Burger	0.021	0.244
book_table	0.029	0.238
online_order	0.019	0.236
DL_Fri	0.024	0.225
Cuis_Caf	0.017	0.217
Cuis_SI	0.019	0.204
Cuis_NI	0.020	0.192
Cuis_CH	0.017	0.189
Cuis_FA	0.020	0.179
DL_Masala	0.018	0.175
DL_Biryani	0.021	0.162
DL_Chicken	0.025	0.104
listed_in(type)	0.017	0.076

Figure E.5

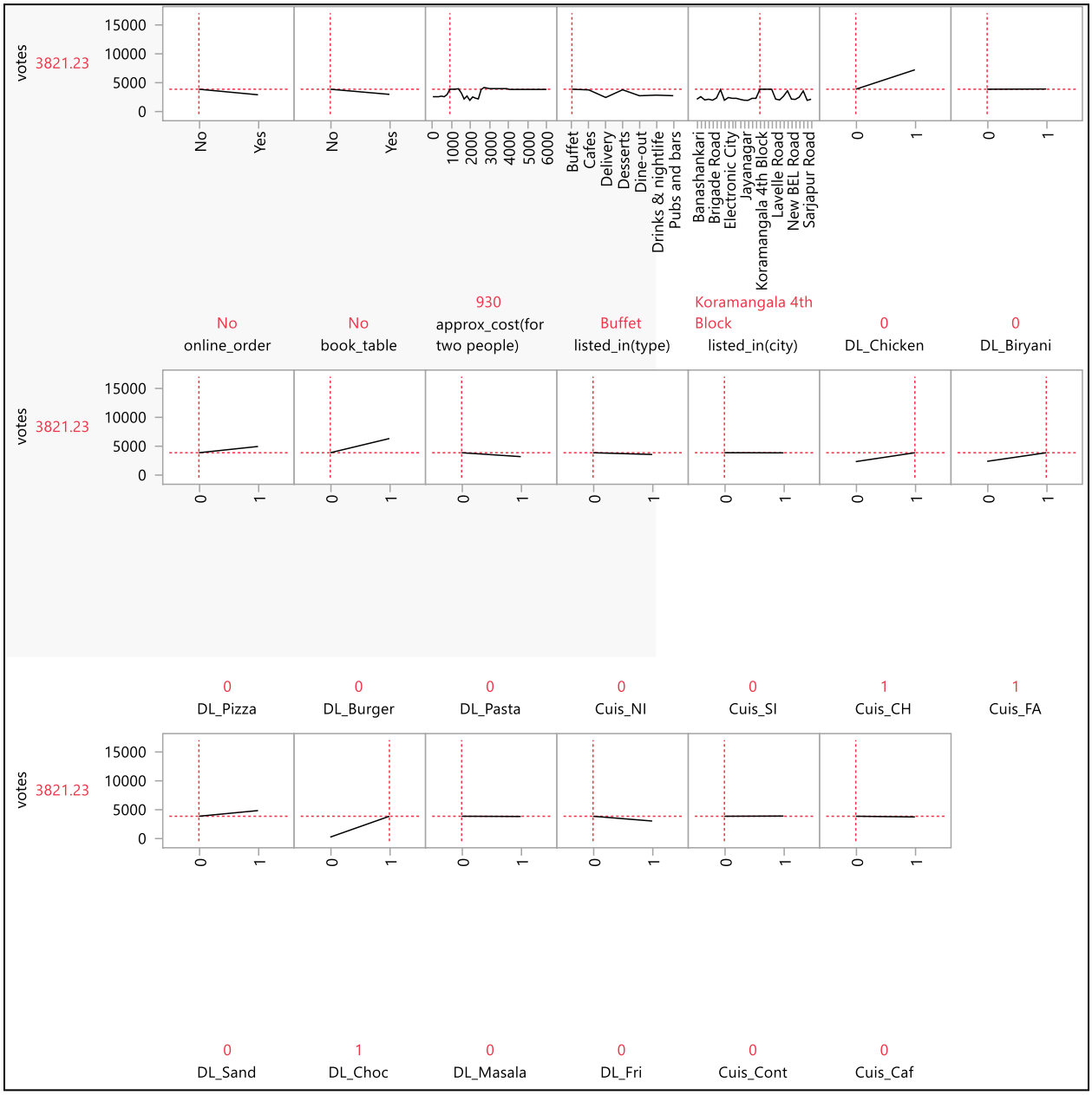
Variable importance analysis of “votes” shows that approximate cost leads the pack in both of the selected models. One difference between the Boosted Tree and Neural Net models, however, is the degree to which approximate cost predicts vote counts. The Boosted Tree model ascribes a 36.4% total effect to approximate cost, while the Neural Net more liberally assigns a total effect of 74.3%.

Another difference between these two models is the ordering of the next top variables. The Boosted Tree model lists *DL_Choc*, *book_table*, and *listed_in(city)* as the next most predictive features of “votes.” Meanwhile, the Neural Net model establishes *listed_in(city)*, *DL_Choc*, and *DL_Sand*(sandwich dish-liked) as the next most important variables.

As with the variable importance figures for “rate,” there is a wide gap between main and total effects here. Not one salient feature is responsible for a restaurant accumulating a large number of votes. It would be unwise for a new restaurant to target a single feature and expect guaranteed success. However, with the correlation between votes and ratings, finding a way to boost vote counts does seem to be a worthwhile endeavor.

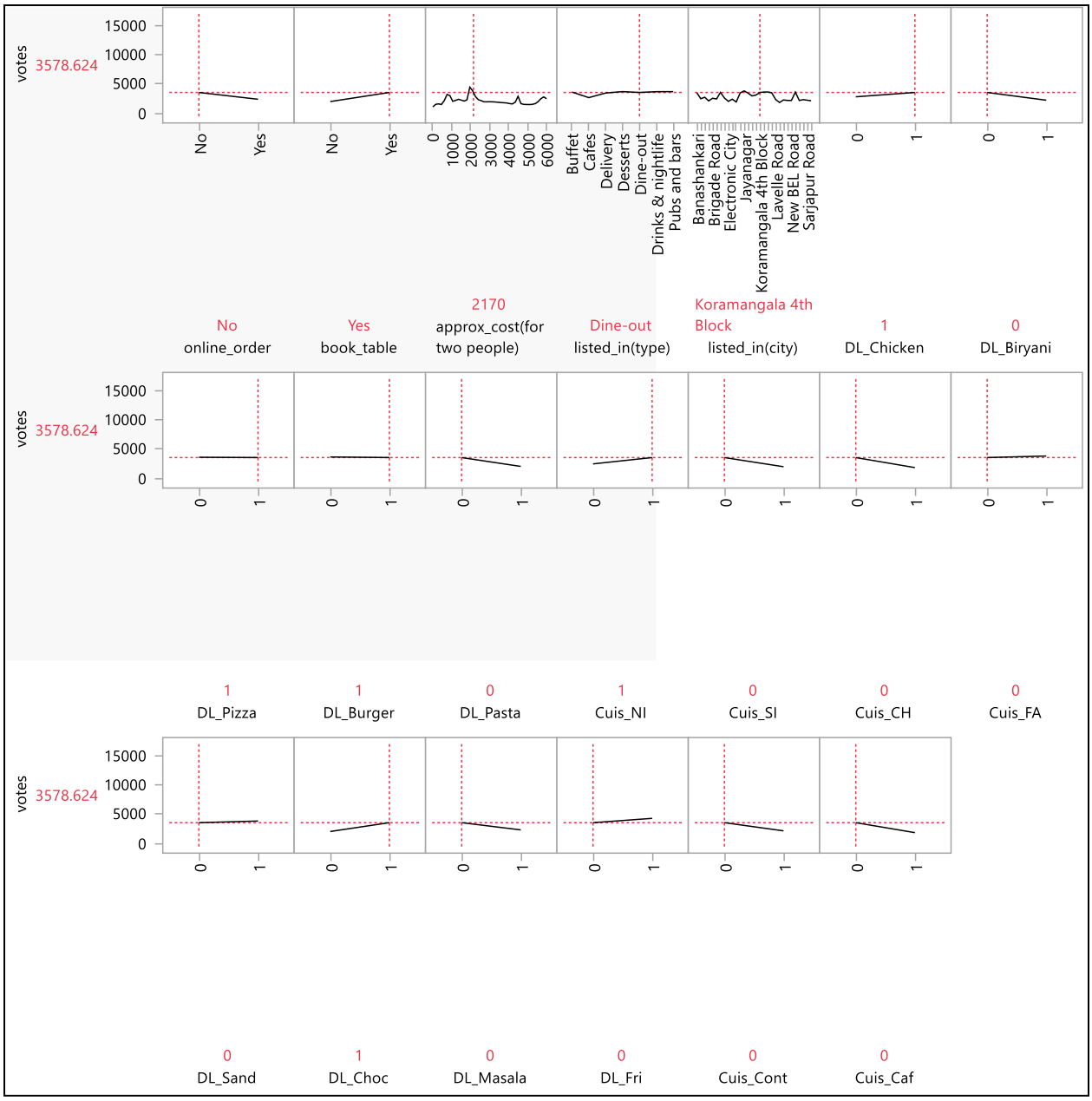
Prediction Profilers - Votes

Boosted Tree - Figure E.6



The Boosted Tree profiler showed very little fluctuation in “vote” for restaurants in the low-to-average cost tier. Only by adjusting “approximate_cost(for two people)” to at least 930 and a handful of binary dish-liked responses to ‘1’ did feature interaction start to become more widespread. It is also interesting to find dips in vote count for certain cost ranges, as well as peaks in vote counts in a handful of city listings.

Neural Net - Figure E.7



The Neural Net profiler also did not reveal much feature interaction with value positions set to their default averages. However, by adjusting certain levels, it becomes easier to discern relationships. Similar to the Boosted Tree, this profiler exposes crests and values in vote counts by approximate cost and city listing. The variables “DL_Choc” and “DL_Sand” interact more noticeably in this profiler. It is interesting to find how the most frequently occurring words in the original “dish_liked” variable seem to bolster “votes.”

Predictive Analytics Review

Through this predictive analytics exercise, we have uncovered appropriate methods for identifying determinants of ratings and vote counts. The selected models do not always guarantee a predicted outcome. And, in fact, the assessments of the models do not always exactly align with each other. However, the models do highlight some of the same key features in their predictions.

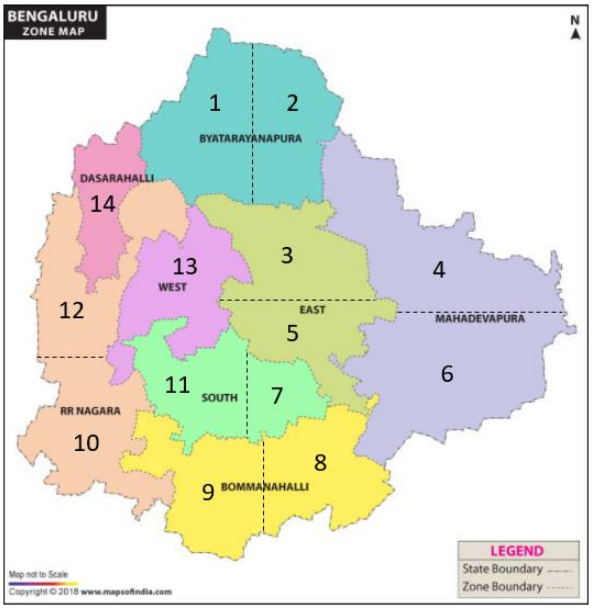
Seeing how features also generally interact within each model provides food for thought for a business looking to enter the Bengaluru restaurant market. Appreciating how ratings and vote counts could potentially rise and fall between features like cost, table bookings, and online orders helps inform conversations surrounding a targeted client base and operational structure. Developing a stronger understanding of variable relationships overall also provides more context for the next prescriptive analytics phase of the project.

F. Prescriptive Model Building

This stage of the analysis will focus on key operational characteristics of features within the Zomato data set. Points of interest include the following: strategic placement of restaurant locations, profitability of key menu items, assignment of delivery orders, and decision-making for a proposed rate/votes promotion. Since not all of the information necessary for prescriptive model building is contained within the data source itself, each model will rely on a series of assumptions. These assumptions will be outlined in the parameters for each problem. The examples in this analysis, conducted in Microsoft Excel, are intended to guide discussions for a new business entering this competitive arena.

Restaurant Location Placement

With an active and vibrant restaurant market in Bengaluru, a key decision for any new business is determining where to position its locations. Opening too few locations would struggle to serve the entire area, while having too many close together could result in proximal locations interfering with the success of others. Determining the appropriate metrics for the number and placement of locations is a complex task that can depend on historical as well as cross-sectional data. We will start by reviewing a map of central Bengaluru:



Retrieved from https://www.mapsofindia.com/maps/karnataka/bangalore-city-zone-map.html

Figure F.1

Some localities in this map have been sub-divided to achieve a more feasible coverage area. We will assume that a new restaurant would like to open enough locations so that each can serve its own area and all of the adjacent sub-localities. The goal of this analysis is to determine the minimum number and proper placement of restaurants to meet this criteria. The formulation of this problem is shown below:

Variables ***Definitions***

x_j { 1 if restaurant is placed in sub-locality *j*, 0 otherwise }

Parameters

v_{ij} { 1 if sub-locality *i* is self-serving or is adjacent to sub-locality *j*, 0 otherwise,
where $i \in \{L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9, L_{10}, L_{11}, L_{12}, L_{13}, L_{14}\}$
and $j \in \{L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9, L_{10}, L_{11}, L_{12}, L_{13}, L_{14}\}\}$

Objective: Minimize: $\sum_j x_j$

Constraints

$\sum (j)\, v_{ij} x_i \geq 1$

$x_{ij} \in \{0, 1\}$

$i \in \{L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9, L_{10}, L_{11}, L_{12}, L_{13}, L_{14}\}$

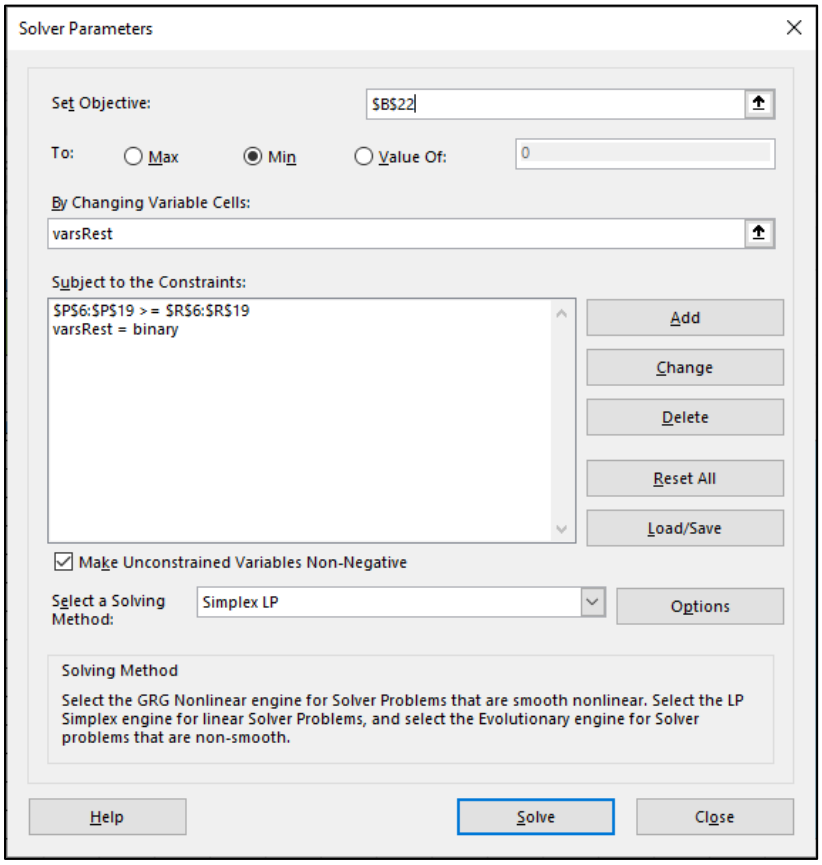
$j \in \{L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9, L_{10}, L_{11}, L_{12}, L_{13}, L_{14}\}$

Figure F.2

Procedure

In order to complete this task, it is necessary to determine which sub-localities share borders with each other. This information has been entered into a matrix in Excel, with a "1" indicating the proposed service area (its own area or adjacency with another sub-locality) and a "0" indicating otherwise. A row for binary variable inputs was created and named as "varsRest." For constraints, the Left-Hand Side of the equation took the sum of the product of each sub-locality's field value multiplied by its corresponding variable field. Each of these product sums has been constrained to being greater than or equal to 1 on the Right-Hand Side. The objective field simply represents the sum of "1" responses in the optimized solution.

The Solver Tool in Excel was deployed, using the following Parameters:



The objective field was set to be minimized by changing the "varsRest" array. The constraint formulations were saved, with an additional indication that the "varsRest" array contains binary values. Simplex LP was selected as the Solving Method. The solution is shown below:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	**Objective** (Min Locations)		
2	Restaurant Location (1=Yes, 0=No)	1	0	0	0	1	0	0	0	1	0	0	0	0	0	3		
3																		
4	**Constraints:**																	
5		L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	LHS	Sign	RHS
6	L1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	>=	1
7	L2	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	>=	1
8	L3	1	1	1	1	1	0	0	0	0	0	0	0	1	0	2	>=	1
9	L4	0	1	1	1	1	1	0	0	0	0	0	0	0	0	1	>=	1
10	L5	0	0	1	1	1	1	1	1	0	0	1	0	1	0	1	>=	1
11	L6	0	0	0	1	1	1	0	1	0	0	0	0	0	0	1	>=	1
12	L7	0	0	0	0	1	0	1	1	1	0	1	0	0	0	2	>=	1
13	L8	0	0	0	0	1	1	1	1	1	0	0	0	0	0	2	>=	1
14	L9	0	0	0	0	0	0	1	1	1	1	1	0	0	0	1	>=	1
15	L10	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	>=	1
16	L11	0	0	0	0	1	0	1	0	1	1	1	0	1	0	2	>=	1
17	L12	1	0	0	0	0	0	0	0	0	1	0	1	1	1	1	>=	1
18	L13	1	0	1	0	1	0	0	0	0	1	1	1	1	0	2	>=	1
19	L14	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	>=	1

Figure F.3

This analysis concludes that three locations would satisfy the minimum number of restaurants to service their own area and all adjacent sub-localities. Based on how the Bengaluru localities are presently subdivided, restaurant locations positioned in sub-localities 1, 5, and 9 would achieve this goal. If the map were to be partitioned with a different configuration of service areas, this analysis would need to be reiterated.

Menu Item Profitability

Prescriptive analytics can also be used to determine performance targets for key menu items that would result in the maximum profit. In this problem, we will make the following assumptions for commonly preferred dishes within the low- to mid-range for a 2-person approximate cost:

Selling Prices		Cost of Goods Sold		Daily Supply		Daily Demand	
Biryani	₹ 400	Biryani	₹ 100	Biryani	40 Units	Biryani	50 Units
Sandwich	₹ 350	Sandwich	₹ 100	Sandwich	70 Units	Sandwich	60 Units
Chocolate	₹ 150	Chocolate	₹ 50	Chocolate	25 Units	Chocolate	20 Units
Burger	₹ 400	Burger	₹ 150	Burger	75 Units	Burger	60 Units
Pasta	₹ 400	Pasta	₹ 150	Pasta	50 Units	Pasta	45 Units
Pizza	₹ 300	Pizza	₹ 200	Pizza	40 Units	Pizza	45 Units

Figure F.4

Since the goal of this exercise is to maximize profit, this component should first be calculated by subtracting Cost of Goods Sold from the Selling Price of each item. This is expressed through the coefficients for each variable in the problem formulation's objective function, shown below:

Variables	Definitions
A	Biryani
B	Sandwich
C	Chocolate
D	Burger
E	Pasta
F	Pizza

Objective

Maximize: 300A + 250B + 100C + 250D + 250E + 100F

Constraints

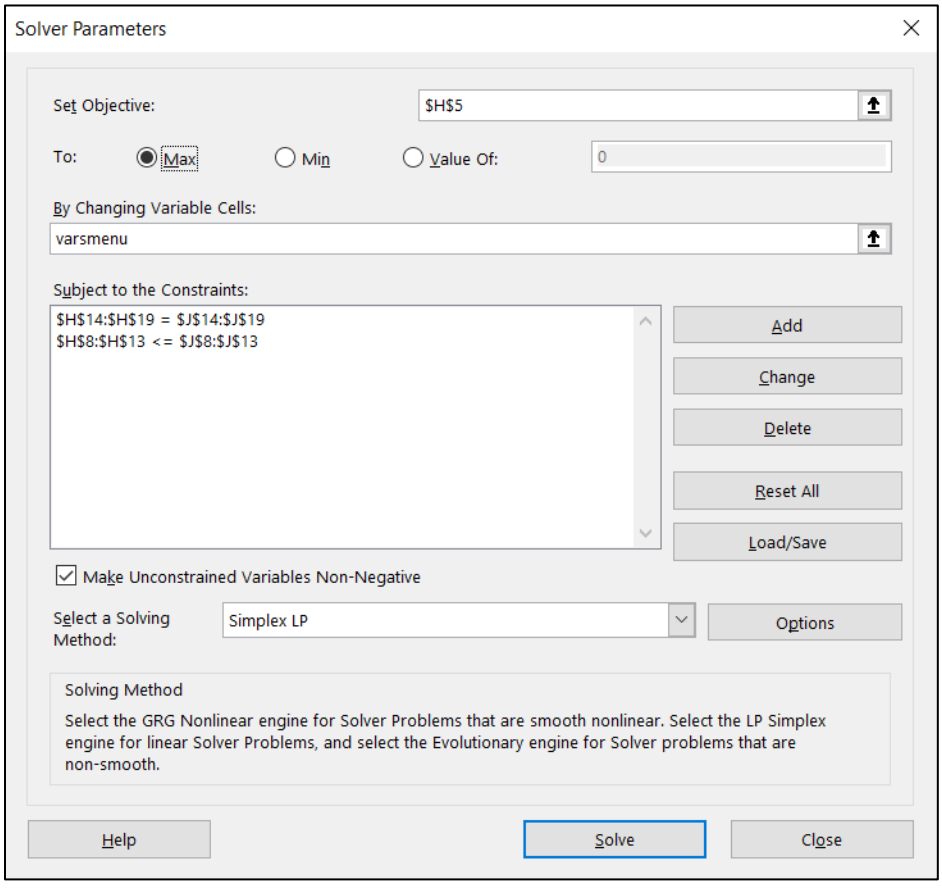
Daily Availability	Daily Demand
A <= 40	A = 50
B <= 70	B = 60
C <= 25	C = 20
D <= 75	D = 60
E <= 50	E = 45
F <= 40	F = 45

Figure F.5

Procedure

This information was entered into Excel. A row for variable inputs was created and named “varsmenu.” A row for each menu item’s profit was added, feeding that figure (multiplied by the proposed number of units to make/sell) into the objective function. The constraints for supply and demand were added, multiplying a coefficient of 1 for each menu item by its variable input. Supply constraints were limited by being less-than or equal to their supply. Demand constraints were set to be equal to the demand of each item.

The Solver Tool parameters are presented below. The objective (Net Profit) is set to be maximized by making changes to the “varsmenu” array of cells. Supply and demand constraints are applied, and Simplex LP is once again selected as the Solving Method.



The solution is presented on the next page.

	A	B	C	D	E	F	G	H	I	J
1		Biryani (A)	Sandwich (B)	Chocolate (C)	Burger (D)	Pasta (E)	Pizza (F)			
2	Items to Make/Sell	40	60	20	60	45	40			
3	Selling Price (rupees)	400	350	150	400	400	300	₹ 94,000		
4	COGS	100	100	50	150	150	200	₹ 34,750		
5	Profit	300	250	100	250	250	100	₹ 59,250	**Objective** (Max Profit)	
6										
7	**Constraints**							**LHS**	**Sign**	**RHS**
8	Biryani Availability	1						40	<=	40
9	Sandwich Availability		1					60	<=	70
10	Chocolate Availability			1				20	<=	25
11	Burger Availability				1			60	<=	75
12	Pasta Availability					1		45	<=	50
13	Pizza Availability						1	40	<=	40
14	Biryani Demand	1						40	=	50
15	Sandwich Demand		1					60	=	60
16	Chocolate Demand			1				20	=	20
17	Burger Demand				1			60	=	60
18	Pasta Demand					1		45	=	45
19	Pizza Demand						1	40	=	45

Figure F.6

With the parameters and assumptions provided, the maximum daily profit would come to 59,250 Rupees. The menu items of *Sandwich*, *Chocolate*, *Burger*, and *Pasta* are all meeting their demand. *Biryani* and *Pizza*, however, have inadequate supply to meet anticipated demand. Analysis results like this may provide justification for a shift in the stock of items like *Biryani* and *Pizza*. Similar analysis could be performed with the supply and demand of any other menu items, and within any other ranges for approximate cost.

Delivery Task Assignment

The next business problem deals with a scenario where a restaurant offers a robust system of online and delivery orders. Orders can be placed in such a way that delivery drivers can be deployed from alternate locations at different times. When multiple orders come through at the same time, the restaurant must determine which drivers should be assigned to which orders. Ideally, this task assignment should be conducted in a way that minimizes the total amount time the entire delivery team is occupied. Every driver should end up with a single assignment. The table below shows the number of minutes each driver would need to fulfill the corresponding order:

	Order 1	**Order 2**	**Order 3**	**Order 4**	**Order 5**
Driver 1	20	8	34	27	15
Driver 2	17	10	12	23	31
Driver 3	25	9	35	28	29
Driver 4	22	10	33	25	30
Driver 5	18	11	30	26	33

Figure F.7

This problem can be formulated as a network flow model. The Drivers will be set as supply nodes, with the Orders as demand nodes.

Variables	Definitions
D	{D_1(Driver 1), D_2(Driver 2), D_3(Driver 3), D_4 (Driver 4), D_5(Driver 5)}
O	{O_1(Order 1), O_2(Order 2), O_3(Order 3), O_4(Order 4), O_5(Order 5)}
X_{ij}	{1 if Driver *i* is assigned to Order *j*, 0 otherwise}
M_{ij}	Total Minutes from *i* to *j*

Objective: Minimize: $\sum(i)\sum(j) M_{ij} X_{ij}$

Constraints

$\sum (j \in O) X_{ij} = 1 \quad \forall \; i \in D$

$\sum (i \in D) X_{ij} = 1 \quad \forall \; j \in O$

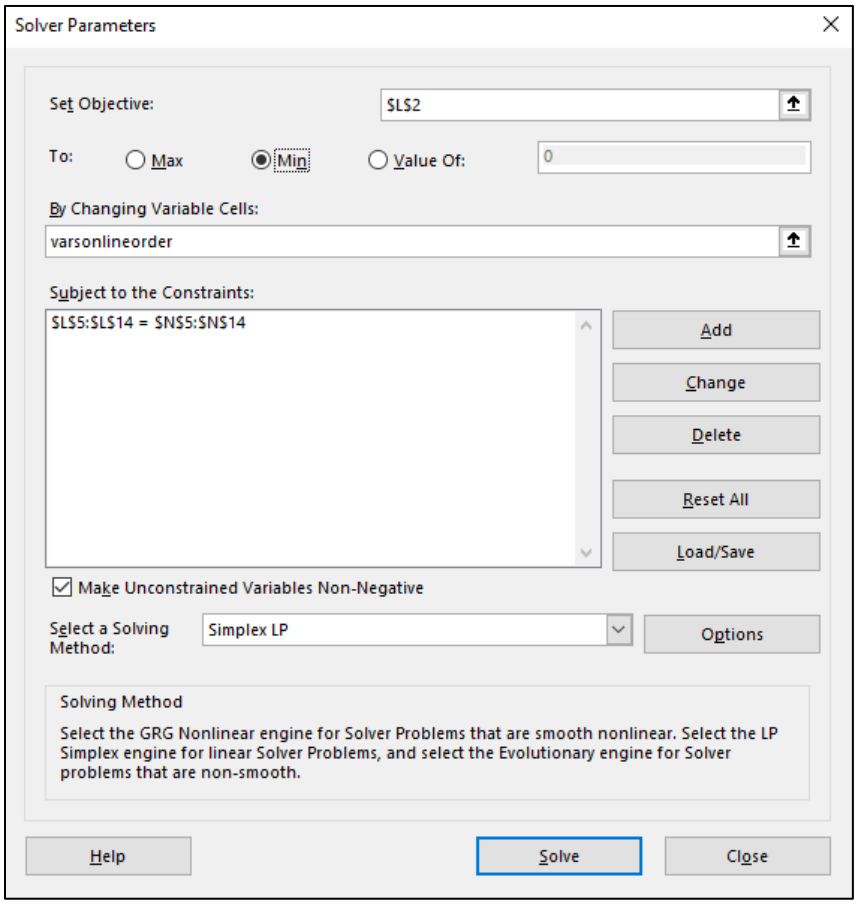
Figure F.8

Procedure

A row for variable inputs has been created, named as “varsonlineorder.” The objective is defined as the sum product of the binary variables in “varsonlineorder” and the total number of minutes elapsed with the routes taken by all Drivers.

Constraints will be represented in a Flow Balance Equation. The supply node flow-ins and demand node flow-outs are set to zero. The flow-outs of supply nodes are set as the sum of the variable values in each Driver’s row. The flow-ins for the demand nodes are then set as the sum of the variable values in each Order’s column. The difference between flow-in and flow-out in each row of the Flow Balance Equation table is fed into the Left-Hand Side of the equation. This constraint should be equal to the Right-Hand Side of the equation, with supply nodes having a ‘-1’ and demand nodes having a ‘1’.

The Solver Tool parameters for this problem are shown below. The objective is set to be minimized by making changes to the “varsonlineorder” array of cells. Supply and demand constraints are applied, and Simplex LP is once again chosen as the Solving Method.



The solution is presented on the next page.

The solution below reveals that 79 minutes is the minimum total amount of time required for all drivers to take their optimal assignments. It is interesting to note that, for the sake of team efficiency, not every driver takes their quickest order. This scenario is heavily dependent upon its framework of assumptions. If orders were placed at different times, or if not all drivers required an assignment, an alternative solution would likely emerge.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1			Orders											
2			O_1	O_2	O_3	O_4	O_5		Objective (Min. Minutes)			79		
3		D_1	20	8	34	27	15							
4		D_2	17	10	12	23	31		Node	Flow in	Flow Ou	LHS	Sign	RHS
5	Drivers	D_3	25	9	35	28	29		1	0	1	-1	=	-1
6		D_4	22	10	33	25	30		2	0	1	-1	=	-1
7		D_5	18	11	30	26	33		3	0	1	-1	=	-1
8			Orders						4	0	1	-1	=	-1
9			O_1	O_2	O_3	O_4	O_5		5	0	1	-1	=	-1
10		D_1	0	0	0	0	1		1	1	0	1	=	1
11		D_2	0	0	1	0	0		2	1	0	1	=	1
12	Drivers	D_3	0	1	0	0	0		3	1	0	1	=	1
13		D_4	0	0	0	1	0		4	1	0	1	=	1
14		D_5	1	0	0	0	0		5	1	0	1	=	1

Figure F.9

Vote Counts Promotion

Earlier in this report, a positive correlation was identified between vote counts and ratings. In this exercise, a restaurant must decide on incentivizing vote counts through a Coupon Book and/or Internet Offers. Estimated return outcomes (net profit/loss generated from higher vote counts) over the duration of the promotion are shown in the table below.

(Returns in Hundreds of ₹)

	Low Outcome	High Outcome
Coupon Book + Internet	Loss of ₹ 200	Gain of ₹ 300
Internet Only	Loss of ₹ 75	Gain of ₹ 200
Do Nothing	Loss of ₹ 0	Gain of ₹ 75

Figure F.10

Procedure

To make a better-informed decision, this problem can be analyzed using a number of different payoff and regret criteria: Maximax, Maximin, Equally Likely, Hurwicz, and Minimax. These criteria align with the general risk tolerance profile of decision makers. The results of this analysis are displayed below:

	A	B	C	D	E	F	G	H	I	J	K
1										α =	0.45
2	**PAYOFFS**	**Outcomes**		**Maximax**		**Maximin**		**Equally Likely**		**Hurwicz**	
3	**Alternatives**	Low	High	Maximum	Choice	Minimum	Choice	Average	Choice	Realism	Choice
4	Coupon Book + Internet	-₹ 200	₹ 300	₹ 300	**Best**	-₹ 200		₹ 50		₹ 25	
5	Internet Only	-₹ 75	₹ 200	₹ 200		-₹ 75		₹ 63	**Best**	₹ 49	**Best**
6	Do Nothing	₹ 0	₹ 75	₹ 75		₹ 0	**Best**	₹ 38		₹ 34	
7	Max	₹ 0	₹ 300								
8											
9	**REGRET**	**Outcomes**		**Minimax**							
10	**Alternatives**	Low	High	Maximum	Choice						
11	Coupon Book + Internet	₹ 200	₹ 0	₹ 200							
12	Internet Only	₹ 75	₹ 100	₹ 100	**Best**						
13	Do Nothing	₹ 0	₹ 225	₹ 225							

Figure F.11

Maximax: This criteria is used to select the option with the maximum payoff in the best-case scenario. The "Coupon Book + Internet" is the clear choice here.

Maximin: This criteria is used to find the option with the maximum payoff in the worst-case scenario. This relates to the "Do Nothing" option.

Equally Likely: Assuming low and high outcomes are equally likely to occur, this criteria is used for choosing the option with the highest average payoff. In this case, that would be "Internet Only."

Hurwicz, Alpha = 0.45: This approach applies a coefficient of realism to a combination of the high and low outcomes. With an alpha value of 0.45 applied, the "Internet Only" option would be selected.

Regret Minimax: Minimizing the opportunity loss, this criteria helps with selecting the option with the lowest maximum regret (once again, "Internet Only").

While the decision to invest in the promotion to incentivize vote counts ultimately lies in the hands of the decision makers, this analysis provides a data-driven rationale to help justify the ultimate choice. Individuals with a high-risk tolerance might be willing to invest in both the coupon book and internet offers, and risk-averse individuals may choose to do nothing. However, three of the criteria reviewed here suggest that investing in the "Internet Only" option would have the greatest chance of success.

Prescriptive Analytics Review

The problems in this stage of the report highlighted a handful of optimization scenarios that can be useful for a business entering the restaurant market in Bengaluru. This type of analysis can be effective at any stage - from determining the placement of locations and adequate stocking levels, to managing day-to-day operations, to marketing for business growth. Relying on data-driven solutions will allow a new restaurant to align its business practices with measurable indicators and benchmarks.

Each new venture can provide its own series of contextual challenges. This makes it necessary to re-evaluate and reconsider key assumptions based on the unique circumstances of each situation. However, this report provides a generalized analytical framework and method that can be used to support a wide range of impactful business decisions.

G. Recommendations and Decision Analysis

The goal of this project was to analyze trends in the Bengaluru restaurant market, in order to identify variables predictive of success for a new business looking to enter this field. Understanding how currently high-performing restaurants structure their locations, menu, and delivery of service can better inform a new venture. The Zomato analysis outlined within this report has managed to provide substantial insight into popular features of Bengaluru restaurants.

The data set provided was reviewed and studied in a number of different ways. The data was first cleaned and prepared for further consideration. Next, the records were scanned for overall traits and characteristics before isolating stratified and paired relationships. Once these associations were identified, it was necessary to determine the degree to which features were correlated and impactful on restaurant success. The final stage of the analysis included a series of exercises that can be used to assist a new restaurant with planning its operations.

Driven and justified by the analysis in each stage, the recommendations of this report pertain to the following decision points:

1. Restaurant Locations and Placement
2. Cuisines Offered
3. Restaurant Pricing and Type
4. Structured Services (Online Ordering, Table Bookings)
5. Opportunities for Further Analysis

Each of these recommendations will be offered with consideration of the analyses performed and the impact on success of the business.

Restaurant Locations and Placement

The descriptive and predictive analytics stages of this report revealed that location is critical to a restaurant's popularity. The locations of **Koramangala**, **Indiranagar**, **Church Street**, and **BTM** appear in the Top 10 list of restaurant vote counts and have a high average rating (*Figure C.12)*. Placing a restaurant in any of these areas would make it likely to be visited and reviewed. Additional factors from the data set, namely pricing, should inform a restaurant's placement as well. Koramangala itself has multiple blocks in the middle price range that may justify more than one location.

The number of locations was explored in the prescriptive analytics section of this report (*Figures F.1 - F.3*). Depending on the available capital and appetite for expansion, the optimum number and placement of locations can be determined by reiterating the analysis performed here. As an example, this analysis showed that as few as three locations may be able to service the entire area. However, if the business would like to invest more aggressively, more locations should be considered.

The decision to open multiple restaurant locations will surely rely on additional studies from this market. A longitudinal view of restaurant popularity is missing from this data set. Limited to a snapshot of cumulative vote counts and aggregated ratings, it is impossible to extrapolate any time series trends. **Further research in this area is recommended**, especially after each new location opens.

Cuisines Offered

The descriptive and predictive analysis in this report also exposed some useful information regarding popular dishes and menu items. The menu items of chicken, biryani, pizza, burger, and pasta were commonly preferred by reviewers (*Figure C.2*). Having sandwiches and chocolate on the menu was predictive of having a higher vote tally (*Figure E.5*). North Indian, South Indian, Chinese cuisines and fast food were also frequently mentioned in reviews (*Figure C.4*). It is tempting to find a way to offer these types of food.

One risk of focusing on supposedly popular dishes, however, is that it may create an unfocused and ambiguous menu or client experience. Buffet-type restaurants would circumvent this issue, by having a wide array of menu items integrated in its offerings. Another risk of reaching too far in meeting the demand for popular dishes is that quality may be

sacrificed in the pursuit of quantity. It will not help a restaurant's popularity to offer inferior versions of popular dishes preferred at other establishments.

It is recommended that the restaurant **start with offering the most commonly preferred dishes and adjust based on subsequent analysis**. Localized studies that emulate the broader Zomato review experience could provide the business with its own tool for optimizing customer satisfaction. Although testing menu configurations is bound to take a few months, this may help provide the restaurant with a more solid foundation as it enters and expands into the market.

Restaurant Pricing and Type

The recommended cost tier for a new restaurant is in the **mid-range, with an approximate cost for two people between 600 and 1200 Rupees**. This mid-range offers great potential for high ratings. The analysis found that lower cost restaurants are more often associated with lower ratings. And, while the higher cost tier has generally higher ratings, it may be more difficult to enter into that group without seeing historical data behind its high performers (*Figure D.9*).

The mid-range cost tier also includes restaurants that have the highest vote counts. This signals that restaurants in this price range are able to reach many customers, unlike the high-cost tier. The mid-range cost tier also offers the opportunity for a restaurant to serve a variety of cuisines. While this segment of the market may seem over-saturated, there are ways to distinguish a new restaurant through its menu items and structured services.

Successful restaurants in the mid-range price category include dine-out and delivery service types as well as a fast casual and/or buffet element for their clients (*Figure D.11*). This suggests the possibility of a time-limited buffet option, such as during lunch-time hours. Agility and flexibility are key characteristics of a successful restaurant in this space. It will be beneficial to **conduct more research for each unique location**. Monitoring local trends in the area may provide useful insight into the development of future offerings and services.

Structured Services (Online Ordering, Table Bookings)

Whether the restaurant offers online ordering and/or table reservations largely depends on the selected restaurant pricing. The analysis in this report acknowledged patterns in the availability of online ordering and table bookings being tied to approximate cost (*Figure D.3*).

The majority of restaurants in the mid-range cost tier offer online ordering, but do not offer table bookings.

Assuming that a new restaurant resides in this cost segment, **it is recommended that online orders are routinely offered.** Online orders would be helpful for expanding the breadth of area coverage, in addition to supporting a delivery service. The logistics of online ordering present unique challenges, and more research would be helpful in determining the best implementation. For example, a robust online and/or social media presence would seem advantageous, along with providing customers a system of making online payments. A review of market competitors should yield valuable design concepts.

In contrast, **it is recommended that table bookings are *not* normally offered**. The option of offering table bookings can remain open for future consideration, for example, for special occasions or large events. Advertising table bookings on a limited basis may help the restaurant differentiate itself from its peers. Studying market competitors can provide insight into how and when to offer this time-limited service. Keeping this option open also allows the restaurant to pivot, in case of popular demand. However, under normal circumstances, table bookings do not seem necessary.

Opportunities for Further Analysis

Opening a new restaurant or introducing a new service requires extensive research that cannot often be accomplished with a singular cross-sectional data source. Each recommendation in this decision analysis has room for additional research. It will be imperative to continue to monitor the progress of the new restaurant and its larger market. The environment is bound to change and shift with the latest trends and influence from competitors.

An iterative time-series analysis of the restaurant's growth in popularity would be useful in determining any correlation or causality between business decisions and customer sentiment (*Figure F.11*). This analysis would necessitate a fair amount of data collection over at least a few months. However, if a new restaurant continues to incorporate data-driven decisions into its business planning, the likelihood of success will remain high.