part3

Akarsh Gaonkar

2025-01-04

```
# Importing Libraries
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
library(lattice)
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.3.3

## randomForest 4.7-1.2

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(rpart)
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.3.3
```

```r
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-8
```

```r
## Task 1: PREDICTIVE MODELLING

# Load Dataset
data <- read.csv("~/Desktop/data_analysis_restaurant_data/Dataset.csv")

# View top 10 rows of the dataset
head(data, 10)
```

```
##    Restaurant.ID                         Restaurant.Name Country.Code
## 1        6317637                    Le Petit Souffle          162
## 2        6304287                    Izakaya Kikufuji          162
## 3        6300002                Heat - Edsa Shangri-La          162
## 4        6318506                                Ooma          162
## 5        6314302                        Sambo Kojin          162
## 6       18189371                       Din Tai Fung          162
## 7        6300781                         Buffet 101          162
## 8        6301290                            Vikings          162
## 9        6300010 Spiral - Sofitel Philippine Plaza Manila          162
## 10       6314987                           Locavore          162
##                City
## 1       Makati City
## 2       Makati City
## 3  Mandaluyong City
## 4  Mandaluyong City
## 5  Mandaluyong City
## 6  Mandaluyong City
## 7        Pasay City
## 8        Pasay City
## 9        Pasay City
## 10       Pasig City
##                                                                                 Address
## 1              Third Floor, Century City Mall, Kalayaan Avenue, Poblacion, Makati City
## 2                  Little Tokyo, 2277 Chino Roces Avenue, Legaspi Village, Makati City
```

```
## 3                              Edsa Shangri-La, 1 Garden Way, Ortigas, Mandaluyong City
## 4               Third Floor, Mega Fashion Hall, SM Megamall, Ortigas, Mandaluyong City
## 5                     Third Floor, Mega Atrium, SM Megamall, Ortigas, Mandaluyong City
## 6              Ground Floor, Mega Fashion Hall, SM Megamall, Ortigas, Mandaluyong City
## 7  Building K, SM By The Bay, Sunset Boulevard, Mall of Asia Complex (MOA), Pasay City
## 8    Building B, By The Bay, Seaside Boulevard, Mall of Asia Complex (MOA), Pasay City
## 9                Plaza Level, Sofitel Philippine Plaza Manila, CCP Complex, Pasay City
## 10                 Brixton Technology Center, 10 Brixton Street, Kapitolyo, Pasig City
##                                        Locality
## 1        Century City Mall, Poblacion, Makati City
## 2       Little Tokyo, Legaspi Village, Makati City
## 3       Edsa Shangri-La, Ortigas, Mandaluyong City
## 4           SM Megamall, Ortigas, Mandaluyong City
## 5           SM Megamall, Ortigas, Mandaluyong City
## 6           SM Megamall, Ortigas, Mandaluyong City
## 7  SM by the Bay, Mall of Asia Complex, Pasay City
## 8  SM by the Bay, Mall of Asia Complex, Pasay City
## 9      Sofitel Philippine Plaza Manila, Pasay City
## 10                                     Kapitolyo
##                                              Locality.Verbose Longitude
## 1        Century City Mall, Poblacion, Makati City, Makati City  121.0275
## 2       Little Tokyo, Legaspi Village, Makati City, Makati City  121.0141
## 3  Edsa Shangri-La, Ortigas, Mandaluyong City, Mandaluyong City  121.0568
## 4      SM Megamall, Ortigas, Mandaluyong City, Mandaluyong City  121.0565
## 5      SM Megamall, Ortigas, Mandaluyong City, Mandaluyong City  121.0575
## 6      SM Megamall, Ortigas, Mandaluyong City, Mandaluyong City  121.0563
## 7   SM by the Bay, Mall of Asia Complex, Pasay City, Pasay City  120.9797
## 8   SM by the Bay, Mall of Asia Complex, Pasay City, Pasay City  120.9793
## 9       Sofitel Philippine Plaza Manila, Pasay City, Pasay City  120.9801
## 10                                     Kapitolyo, Pasig City  121.0565
##    Latitude                          Cuisines Average.Cost.for.two
## 1  14.56544         French, Japanese, Desserts                1100
## 2  14.55371                           Japanese                1200
## 3  14.58140   Seafood, Asian, Filipino, Indian                4000
## 4  14.58532                    Japanese, Sushi                1500
## 5  14.58445                   Japanese, Korean                1500
## 6  14.58376                            Chinese                1000
## 7  14.53133                     Asian, European                2000
## 8  14.54000 Seafood, Filipino, Asian, European                2000
## 9  14.55299            European, Asian, Indian                6000
## 10 14.57204                           Filipino                1100
##           Currency Has.Table.booking Has.Online.delivery Is.delivering.now
## 1  Botswana Pula(P)               Yes                  No                No
## 2  Botswana Pula(P)               Yes                  No                No
## 3  Botswana Pula(P)               Yes                  No                No
## 4  Botswana Pula(P)                No                  No                No
## 5  Botswana Pula(P)               Yes                  No                No
## 6  Botswana Pula(P)                No                  No                No
## 7  Botswana Pula(P)               Yes                  No                No
## 8  Botswana Pula(P)               Yes                  No                No
## 9  Botswana Pula(P)               Yes                  No                No
## 10 Botswana Pula(P)               Yes                  No                No
##    Switch.to.order.menu Price.range Aggregate.rating Rating.color Rating.text
## 1                    No           3              4.8   Dark Green   Excellent
```

```
## 2                    No           3              4.5   Dark Green   Excellent
## 3                    No           4              4.4        Green   Very Good
## 4                    No           4              4.9   Dark Green   Excellent
## 5                    No           4              4.8   Dark Green   Excellent
## 6                    No           3              4.4        Green   Very Good
## 7                    No           4              4.0        Green   Very Good
## 8                    No           4              4.2        Green   Very Good
## 9                    No           4              4.9   Dark Green   Excellent
## 10                   No           3              4.8   Dark Green   Excellent
##    Votes
## 1    314
## 2    591
## 3    270
## 4    365
## 5    229
## 6    336
## 7    520
## 8    677
## 9    621
## 10   532
```

```r
# Build a regression model to predict the aggregate rating of a restaurant based
# on available features
# Split the dataset into training and testing sets and evaluate the model's
#  performance using appropriate metrics.
# Experiment with different algorithms (e.g., linear regression, decision trees,
#. random forest) and compare their performance

# Create new numerical columns
data$Has.Table.Booking_Num <- ifelse(data$Has.Table.booking == "Yes", 1, 0)
data$Has.Online.Delivery_Num <- ifelse(data$Has.Online.delivery == "Yes", 1, 0)
```

```r
# Split data into training and testing sets
set.seed(123) # for reproducibility
trainIndex <- createDataPartition(data$Aggregate.rating, p = 0.8, list = FALSE, times = 1)
data_train <- data[trainIndex, ]
data_test  <- data[-trainIndex, ]
```

```r
# Define predictor variables and target variable
predictors <- c("Average.Cost.for.two", "Votes", "Price.range", "Has.Table.Booking_Num", "Has.Online.Del
target_variable <- "Aggregate.rating"
```

```r
# Linear Regression
lm_model <- train(as.formula(paste(target_variable, "~", paste(predictors,
                                                               collapse = " + "))), data = data_train,
                  method = "lm",
                  trControl = trainControl(method = "cv"))

# Decision Tree Regression
tree_model <- train(as.formula(paste(target_variable, "~", paste(predictors, collapse = " + "))), data =
                    method = "rpart",
                    trControl = trainControl(method = "cv"))
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
```

```
## : There were missing values in resampled performance measures.
```

```
# Random Forest Regression
rf_model <- train(as.formula(paste(target_variable, "~", paste(predictors, collapse = " + "))), data = d
                  method = "rf",
                  trControl = trainControl(method = "cv"))
```

```
# Predictions on test set
lm_pred <- predict(lm_model, newdata = data_test)
tree_pred <- predict(tree_model, newdata = data_test)
rf_pred <- predict(rf_model, newdata = data_test)
```

```
# Evaluation
lm_rmse <- sqrt(mean((lm_pred - data_test$Aggregate.rating)^2))
lm_r2 <- cor(lm_pred, data_test$Aggregate.rating)^2

tree_rmse <- sqrt(mean((tree_pred - data_test$Aggregate.rating)^2))
tree_r2 <- cor(tree_pred, data_test$Aggregate.rating)^2

rf_rmse <- sqrt(mean((rf_pred - data_test$Aggregate.rating)^2))
rf_r2 <- cor(rf_pred, data_test$Aggregate.rating)^2
```

```
# Print results
cat("Linear Regression RMSE:", lm_rmse, "\n")
```

```
## Linear Regression RMSE: 1.294572
```

```
cat("Linear Regression R-squared:", lm_r2, "\n\n")
```

```
## Linear Regression R-squared: 0.267201
```

```
cat("Decision Tree RMSE:", tree_rmse, "\n")
```

```
## Decision Tree RMSE: 0.3984173
```

```
cat("Decision Tree R-squared:", tree_r2, "\n\n")
```

```
## Decision Tree R-squared: 0.9308097
```

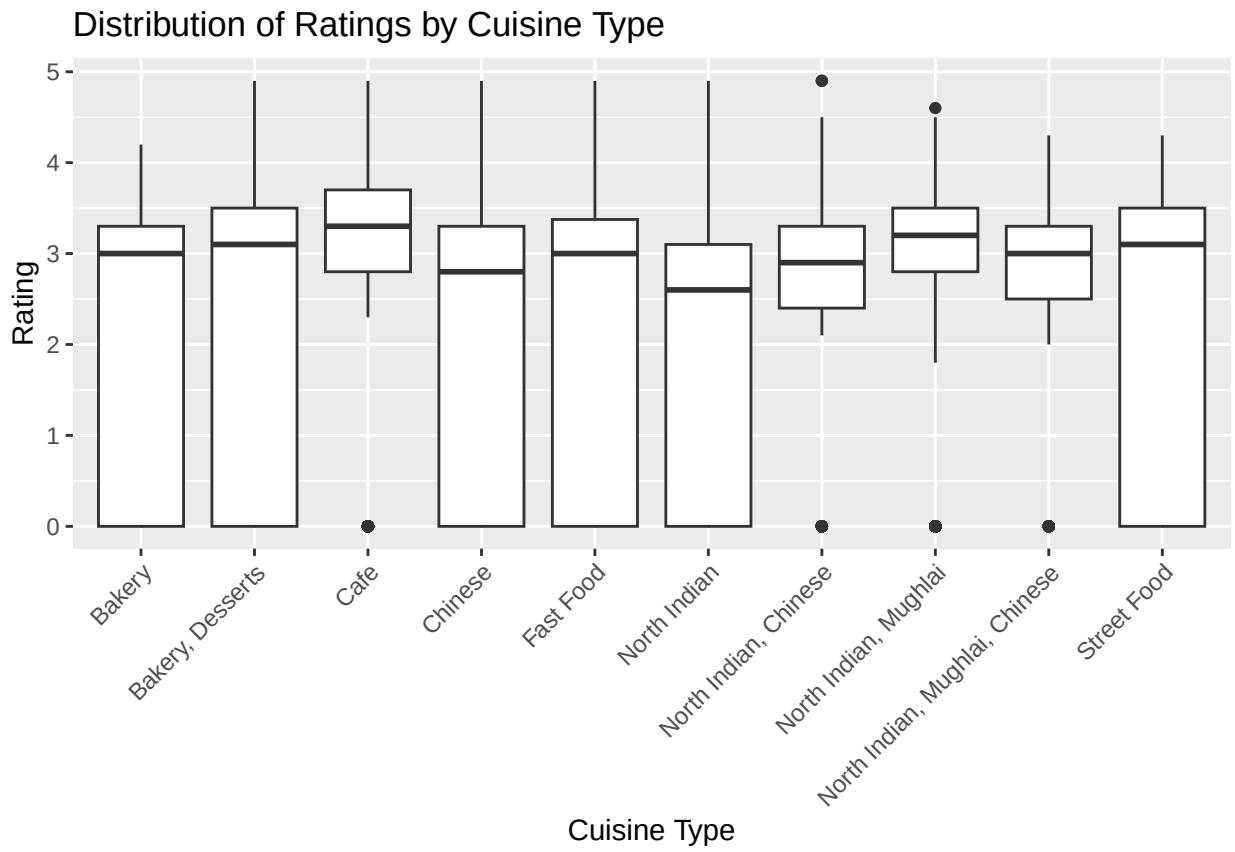
```
cat("Random Forest RMSE:", rf_rmse, "\n")
```

```
## Random Forest RMSE: 0.3370269
```

```
cat("Random Forest R-squared:", rf_r2, "\n")
```

```
## Random Forest R-squared: 0.9503985
```

```r
## Task 2: CUSTOMER PREFERENCE ANALYSIS


# Analyze the relationship between the type of cuisine and the restaurant's
#. rating

# Identify the top 10 cuisines
top_cuisines <- head(sort(table(data$Cuisines), decreasing = TRUE), 10)
top_cuisines <- names(top_cuisines)

# Subset the data for only the top 10 cuisines
data_top_cuisines <- data[data$Cuisines %in% top_cuisines, ]

# Create a box plot to visualize the distribution of ratings for each cuisine
#  type (top 10)
ggplot(data_top_cuisines, aes(x = Cuisines, y = Aggregate.rating)) +
  geom_boxplot() +
  labs(x = "Cuisine Type", y = "Rating", title = "Distribution of Ratings by Cuisine Type") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```r
# Identify the most popular cuisines among customers based on the number of votes

# Group the data by cuisine and calculate the total number of votes for each
# cuisine
popular_cuisines <- data %>%
```

```
  group_by(Cuisines) %>%
  summarise(TotalVotes = sum(Votes, na.rm = TRUE)) %>%
  arrange(desc(TotalVotes))

# Print the top 10 most popular cuisines
head(popular_cuisines, 10)
```

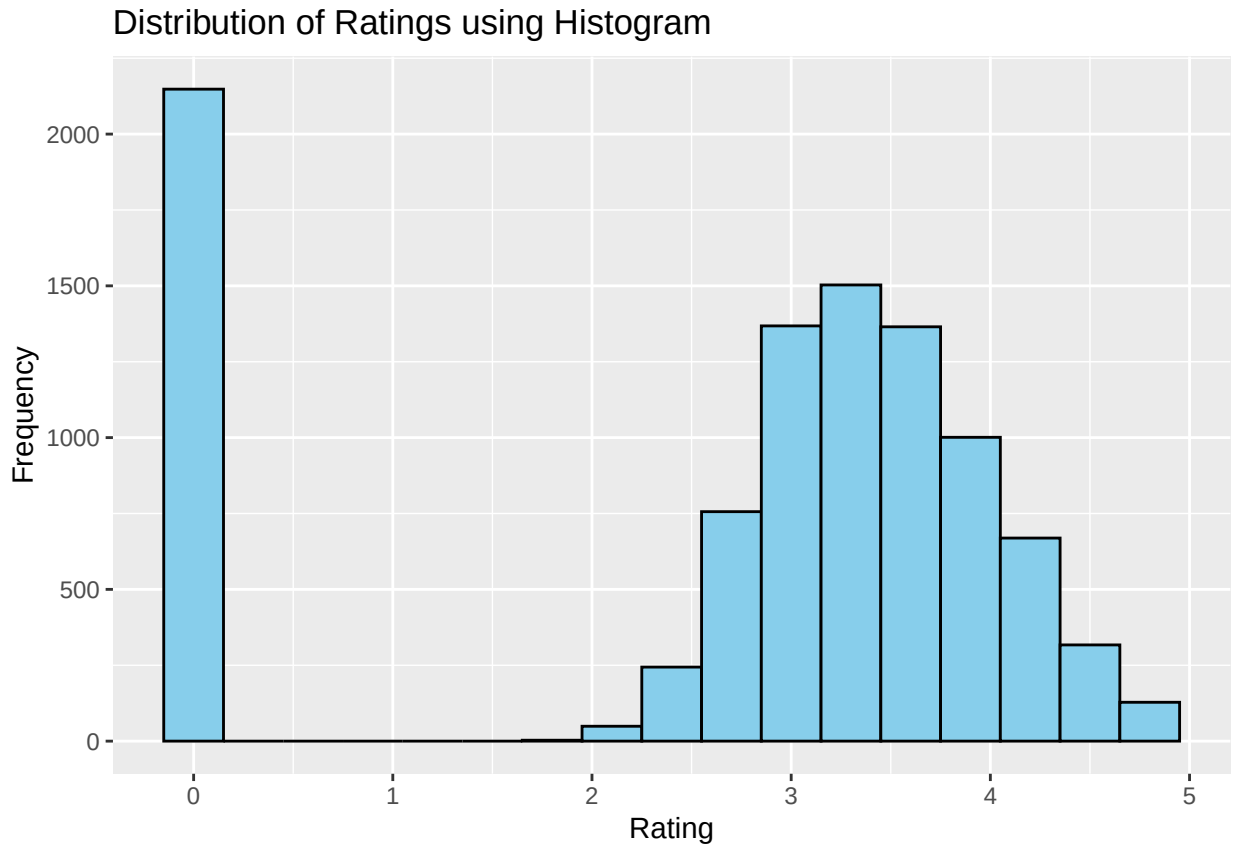
```
## # A tibble: 10 x 2
##    Cuisines                        TotalVotes
##    <chr>                                <int>
##  1 North Indian, Mughlai                53747
##  2 North Indian                         46241
##  3 North Indian, Chinese                42012
##  4 Cafe                                 30657
##  5 Chinese                              21925
##  6 North Indian, Mughlai, Chinese       20115
##  7 Fast Food                            17852
##  8 South Indian                         16433
##  9 Mughlai, North Indian                15275
## 10 Italian                              14799
```

```
# Determine if there are any specific cuisines that tend to receive higher
# ratings

# Calculate the average rating for each cuisine
average_ratings <- data %>%
  group_by(Cuisines) %>%
  summarise(AvgRating = mean(`Aggregate.rating`, na.rm = TRUE)) %>%
  arrange(desc(AvgRating))

# Print the cuisines with the highest average ratings
head(average_ratings, 10)
```

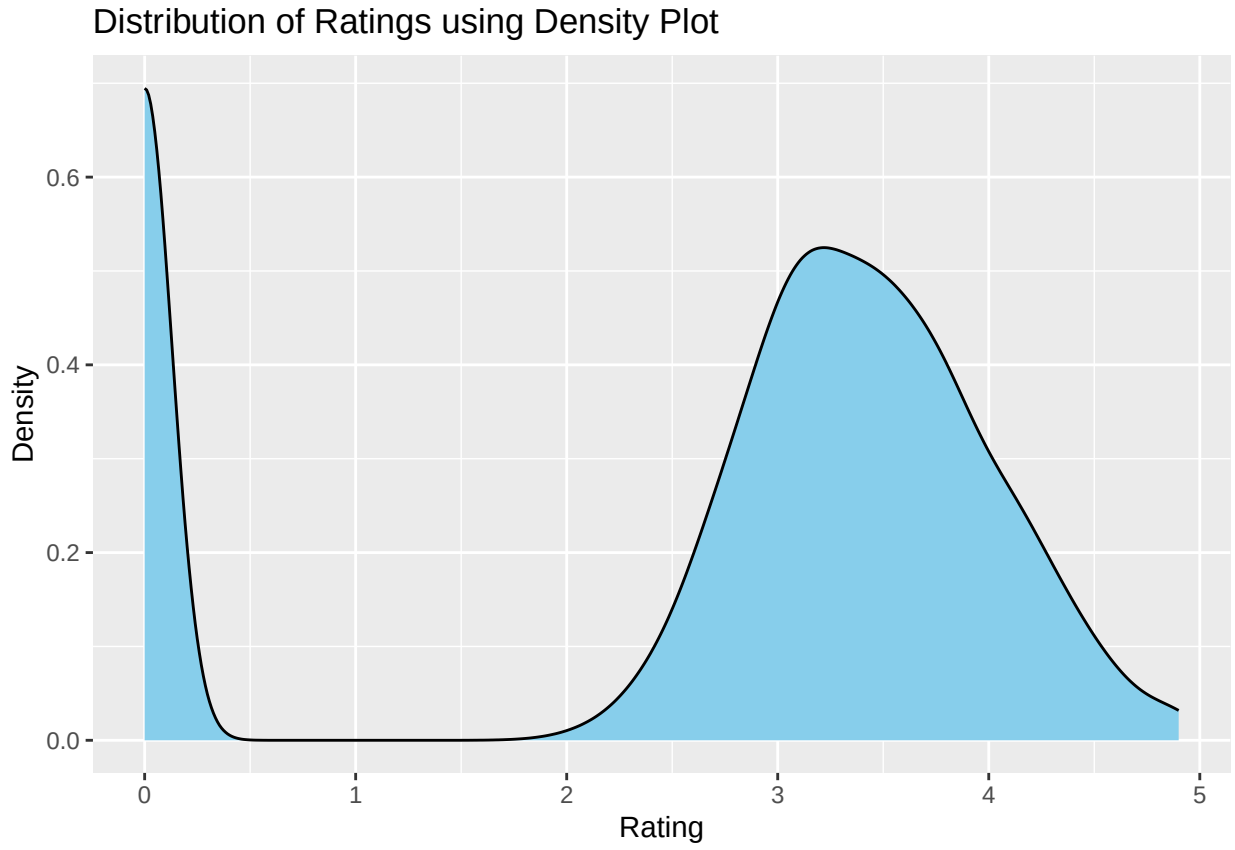
```
## # A tibble: 10 x 2
##    Cuisines                      AvgRating
##    <chr>                             <dbl>
##  1 American, BBQ, Sandwich             4.9
##  2 American, Burger, Grill             4.9
##  3 American, Caribbean, Seafood        4.9
##  4 American, Coffee and Tea            4.9
##  5 American, Sandwich, Tea             4.9
##  6 BBQ, Breakfast, Southern            4.9
##  7 Burger, Bar Food, Steak             4.9
##  8 Continental, Indian                 4.9
##  9 European, Asian, Indian             4.9
## 10 European, Contemporary              4.9
```

```
## Task 3: DATA VISUALIZATION


# Create visualizations to represent the distribution of ratings using different
#. charts (histogram, bar plot, etc.)
```

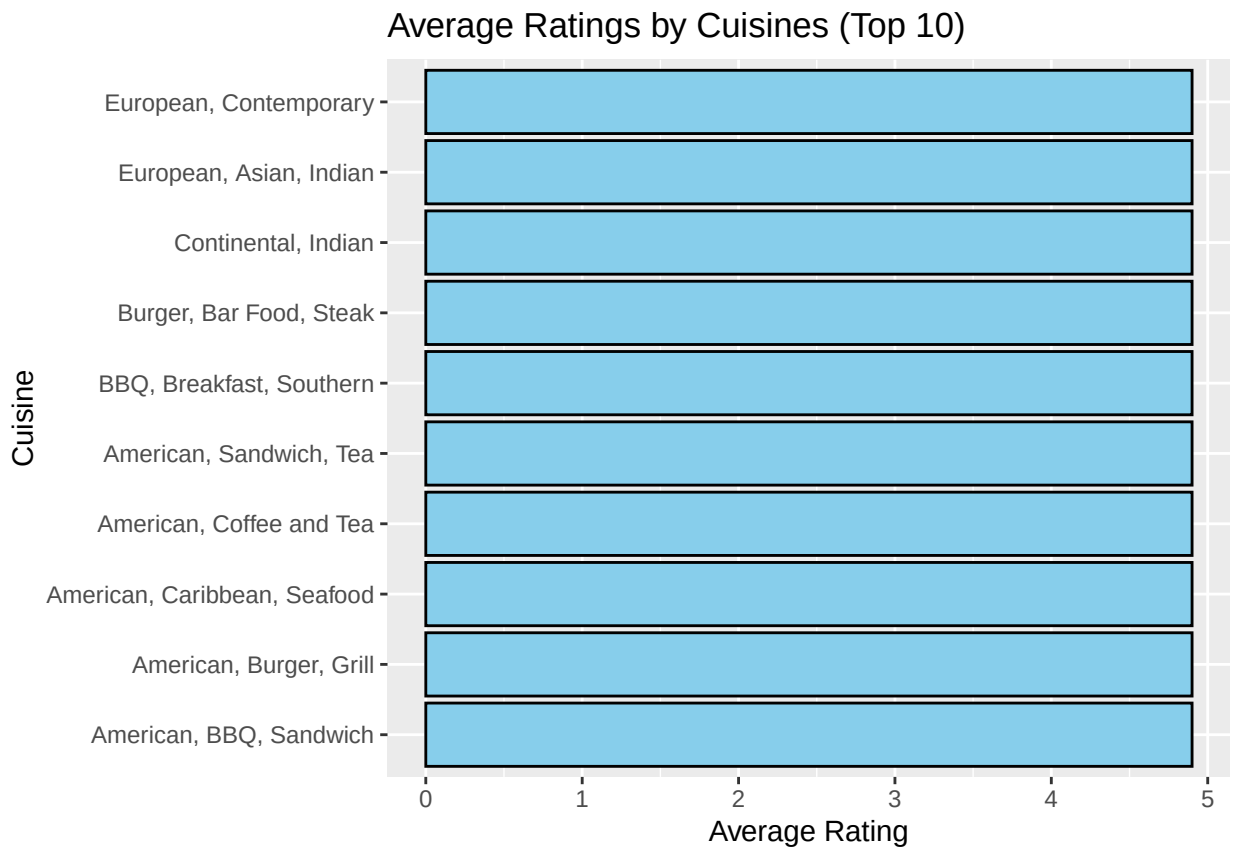
```r
# Create a histogram of ratings
ggplot(data, aes(x = `Aggregate.rating`)) +
  geom_histogram(binwidth = 0.3, fill = "skyblue", color = "black") +
  labs(x = "Rating", y = "Frequency", title = "Distribution of Ratings using Histogram")
```



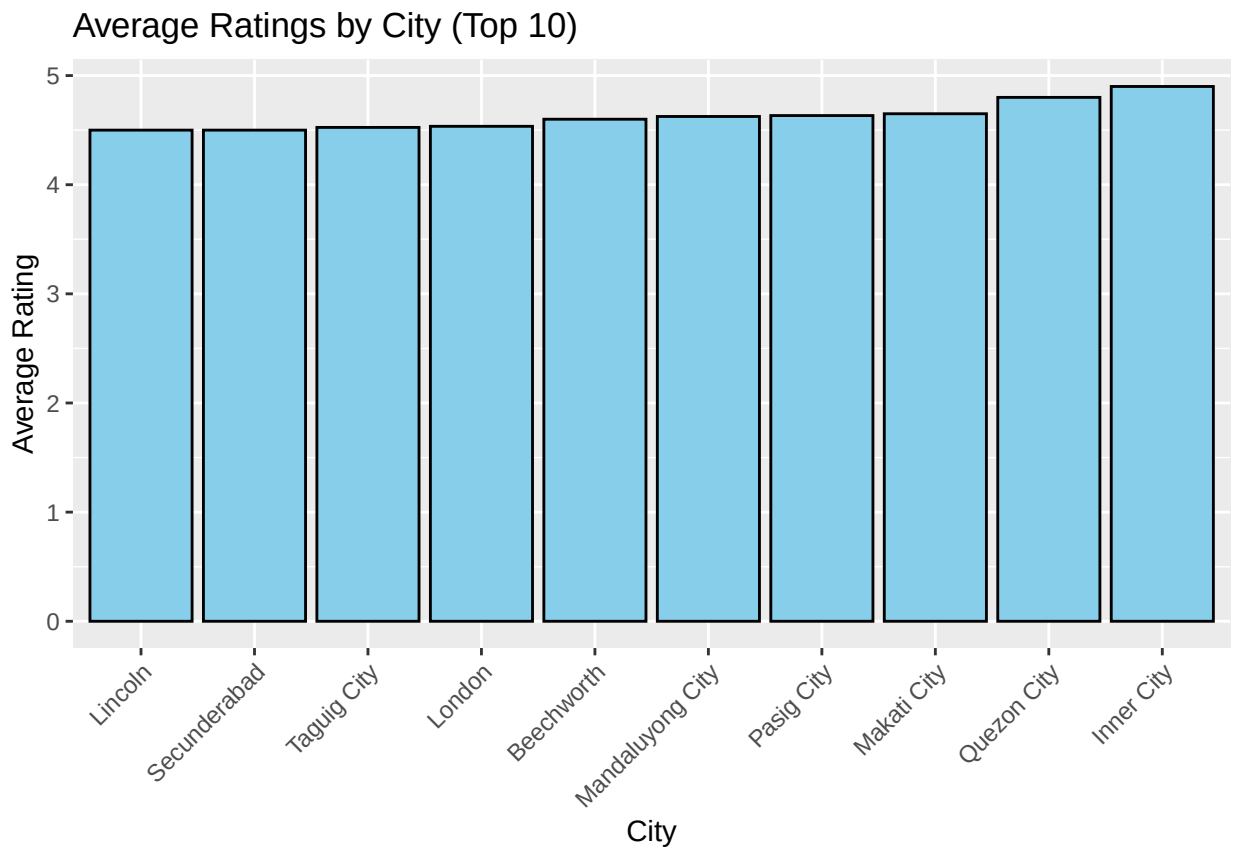

```r
# Create a density plot of ratings
ggplot(data, aes(x = `Aggregate.rating`)) +
  geom_density(fill = "skyblue", color = "black") +
  labs(x = "Rating", y = "Density", title = "Distribution of Ratings using Density Plot")
```



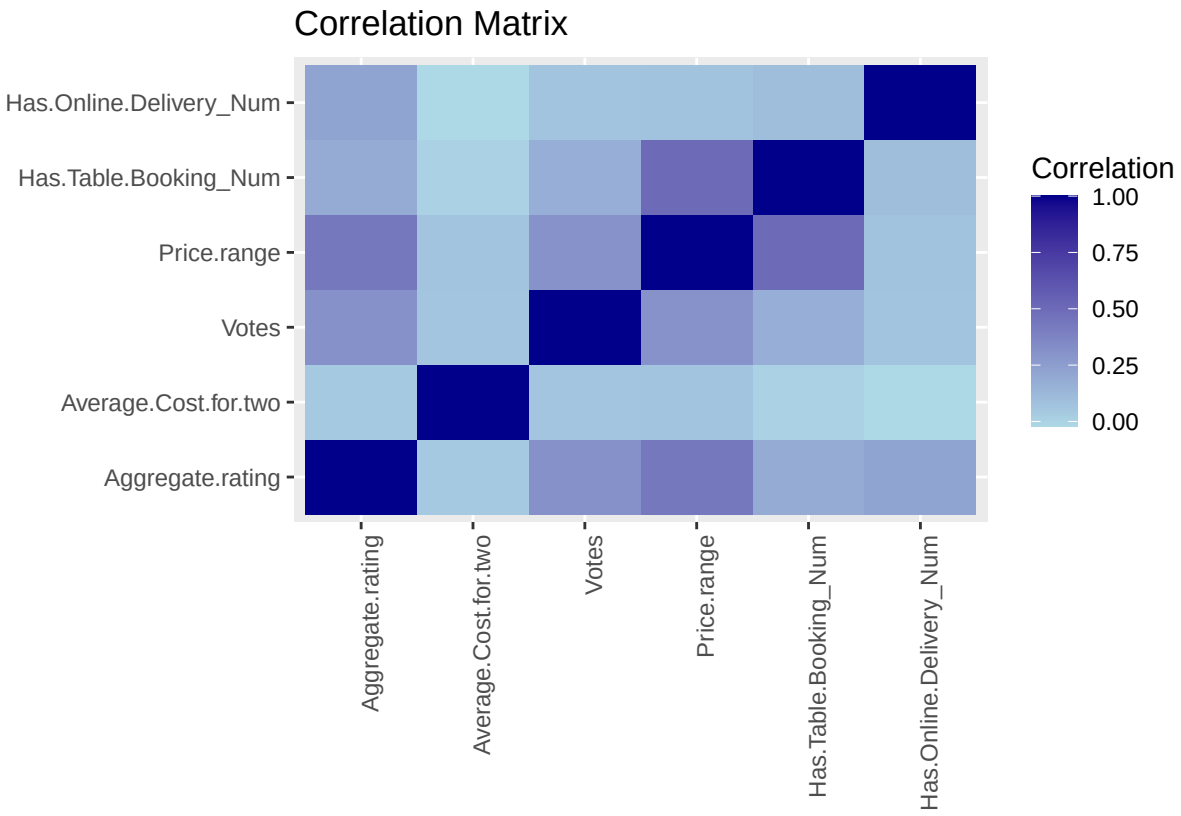

```
# Compare the average ratings of different cuisines or cities using appropriate
#.   visualizations

# Calculate average ratings for each cuisine
average_ratings_cuisine <- data %>%
  group_by(Cuisines) %>%
  summarise(AvgRating = mean(`Aggregate.rating`, na.rm = TRUE)) %>%
  arrange(desc(AvgRating)) %>%
  slice(1:10)

# Create bar plot for average ratings by cuisine
ggplot(average_ratings_cuisine, aes(x = AvgRating, y = reorder(Cuisines, AvgRating))) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  labs(x = "Average Rating", y = "Cuisine", title = "Average Ratings by Cuisines (Top 10)")
```




```r
# Calculate average ratings for each city
average_ratings_city <- data %>%
  group_by(City) %>%
  summarise(AvgRating = mean(`Aggregate.rating`, na.rm = TRUE)) %>%
  arrange(desc(AvgRating)) %>%
  slice(1:10)

# Create bar plot for average ratings by city
ggplot(average_ratings_city, aes(x = reorder(City, AvgRating), y = AvgRating)) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  labs(x = "City", y = "Average Rating", title = "Average Ratings by City (Top 10)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# Visualize the relationship between various features and the target variable to
# gain insights

# Calculate the correlation matrix
correlation_matrix <- cor(data[, c("Aggregate.rating", "Average.Cost.for.two", "Votes", "Price.range",

# Convert correlation matrix to dataframe
correlation_df <- as.data.frame(as.table(correlation_matrix))
names(correlation_df) <- c("Var1", "Var2", "Correlation")

# Visualize the correlation matrix as a heatmap
ggplot(data = correlation_df, aes(x = Var1, y = Var2, fill = Correlation)) +
  geom_tile() +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  labs(title = "Correlation Matrix", x = "", y = "") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        axis.text.y = element_text(angle = 0, hjust = 1))
```


Correlation Matrix
Has.Online.Delivery_Num
Has.Table.Booking_Num
Price.range
Votes
Average.Cost.for.two
Aggregate.rating
Aggregate.rating
Average.Cost.for.two
Votes
Price.range
Has.Table.Booking_Num
Has.Online.Delivery_Num
Correlation
1.00
0.75
0.50
0.25
0.00