1. Data Exploration and Preprocessing

Explore the dataset and identify the number of rows and columns.

## Data Exploration and Preprocessing

# Load Dataset

data <- read.csv("D:/data analysis/Data\_Analytics\_BootCamp 1.0/Dataset.csv")

# View 10 rowa of the dataset

head(data, 10)

# Explore the dataset to indentify the number of rows and columns

cat('Number of rows: ', nrow(data), '\n')

cat('Number of columns: ', ncol(data))

1. Check for missing values in each column and handle them accordingly.

# Check for missing values in each column and handle them accordingly

# Check for missing values

missing\_values <- sum(is.na(data))

# check for empty values

empty\_values <- sum(data == "")

cat('Missing values count: ', missing\_values, '\n')

cat('Empty values count: ', empty\_values)

# checking which columns have the empty values

col\_empty\_values\_count <- colSums(data == '')

cat('Empty values count:\n')

print(col\_empty\_values\_count)

# Drop empty values in the Cuisines column

data <- data[!(data$Cuisines == ''), , drop = FALSE]

# check for empty values

empty\_values <- sum(data == "")

cat('Empty values count: ', empty\_values)

# Check for duplicate

dup <- sum(duplicated(data))

cat('Number of duplicate rows: ', dup)

1. Perform data type conversion if necessary. Analyse the distribution of the target variable ("Aggregate rating") and identify any class imbalances.

# Display basic information about the dataset to check various data types

str(data)

# Analyse the distribution of the target variable ("Aggregate rating") and identify any class imbalances

check\_for\_balance <- summary(data$Aggregate.rating)

print(check\_for\_balance)

# check if the distribution is balance

target\_counts <- table(data$Aggregate.rating)

is\_balanced <- all(target\_counts >= mean(target\_counts))

if(is\_balanced){

print('The distribution of the target variable is balanced.')

} else {

print('The distribution of the target variable is imbalanced.')

}

1. Descriptive Analysis

Calculate basic statistical measures (mean, median, standard deviation, etc.) for numerical columns.

Explore the distribution of categorical variables like "Country Code," "City," and "Cuisines."

Identify the top cuisines and cities with the highest number of restaurants.

## Descriptive Analysis

# Calculate basic statistical measures (mean, median, standard deviation, etc.) for numerical columns

# Select numerical columns

numerical\_columns <- data[, sapply(data, is.numeric)]

# Calculate basic statistical measures

summary\_stats <- summary(numerical\_columns)

print(summary\_stats)

# Calculate the standard deviation for the numerical columns

sds <- sapply(data[, sapply(data, is.numeric)], sd)

print(sds)

# The distribution of categorical variables like Country Code, City, and Cuisines

# Create a count plot for country code

ggplot(data, aes(x = factor(Country.Code))) +

geom\_bar(fill = 'skyblue') +

labs(

title = 'Distribution of Restaurant by Country Codes',

x = 'Country Codes',

y = 'Number of Restaurants'

)

# create a subset of the data containing the top 10 cities

top\_10\_cities <- head(names(sort(table(data$City), decreasing = TRUE)), 10)

data\_top\_10\_cities <- data[data$City %in% top\_10\_cities, ]

# Create count plot for the top 10 cities

ggplot(data = data\_top\_10\_cities, aes(y = factor(City, levels = rev(top\_10\_cities)))) +

geom\_bar(fill = 'steelblue', width = 0.5, stat = 'count') +

labs(

title = 'Top 10 Cities with Highest Number of Restaurants',

x = 'Number of Restaurants',

y = 'Name of Cities'

)

# create a subset of the data containing the top 10 cuisines

top\_10\_cuisines <- head(names(sort(table(data$Cuisines), decreasing = TRUE)), 10)

data\_top\_10\_cuisines <- data[data$Cuisines %in% top\_10\_cuisines, ]

# Create count plot for the top 10 cities

ggplot(data = data\_top\_10\_cuisines, aes(y = factor(Cuisines, levels = rev(top\_10\_cuisines)))) +

geom\_bar(fill = 'steelblue', width = 0.5, stat = 'count') +

labs(

title = 'Top 10 Cuisines with Highest Number of Restaurants',

x = 'Number of Restaurants',

y = 'Name of Cuisines'

)

# The top cuisines and cities with the highest number of restaurants

# Identify the top 10 cuisines and their count

top\_cuisines <- head(sort(table(data$Cuisines), descreasing = TRUE), 10)

# Create a DataFrame with the top 10 cuisine names and count

top\_cuisines\_df <- data.frame(Cuisines = names(top\_cuisines), Count = as.numeric(top\_cuisines))

print('Top 10 Cuisines with the Highest Number of Restaurants:')

print(top\_cuisines\_df)

# Identify the top 10 city and their count

top\_city <- head(sort(table(data$City), descreasing = TRUE), 10)

# Create a DataFrame with the top 10 city names and count

top\_city\_df <- data.frame(City = names(top\_city), Count = as.numeric(top\_city))

print('Top 10 City with the Highest Number of Restaurants:')

print(top\_city\_df)

1. Geospatial Analysis

Visualize the locations of restaurants on a map using latitude and longitude information.

Analyse the distribution of restaurants across different cities or countries. Determine if there is any correlation between the restaurant's location and its rating.

## Geospatial Analysis

# Visualize the locations of restaurants on a map using latitude and longitude information

#create a map of the world

world\_map <- map\_data('world')

# plot restaurant locations on map

ggplot() +

geom\_polygon(data = world\_map, aes(x = long, y = lat, group = group), fill = 'lightgrey', color = 'black') +

geom\_point(data = data, aes(x = Longitude, y = Latitude, color = 'Restaurants'), size = 2) +

scale\_color\_manual(name = 'Legend', values = c(Restaurants = 'red')) +

labs(

title = 'Restaurants Locations on Map',

x = 'Longtitude',

y = 'Latitude'

)

# Analyse the distribution of restaurants across different cities or countries

# create a subset of the data containing the top 10 cities

top\_10\_cities <- head(names(sort(table(data$City), decreasing = TRUE)), 10)

data\_top\_10\_cities <- data[data$City %in% top\_10\_cities, ]

# Create a plot of restaurant across cities

ggplot(data = data\_top\_10\_cities, aes(y = factor(City, levels = rev(top\_10\_cities)))) +

geom\_bar(fill = 'steelblue', width = 0.5, stat = 'count') +

labs(

title = 'Distribution of Restaurant Across Coities',

x = 'Number of Restaurants',

y = 'Name of Cities'

)

# Determine if there is any correlation between the restaurant's location and its rating

# Calculate the correlation

cor\_matrix <- cor(data[c('Longitude', 'Latitude', 'Aggregate.rating')])

# create heatmap to visualize the correlation

corrplot(cor\_matrix, method = 'color', col = colorRampPalette(c('blue', 'white', 'red'))(20),

type = 'upper', order = 'hclust', tl.col = 'black', tl.srt = 45,

title = "Correlation Between Restaurant's Location and Rating",

mar = c(0, 0, 3, 1)

)