Muthukumarasamy.S

2024-06-07

**Level 1**

**Task 1: Data Exploration and Preprocessing**

**Explore the dataset and identify the number of rows and columns. Check for missing values in each column and handle them accordingly. Perform data type conversion if necessary. Analyze the distribution of the target variable (“Aggregate rating”) and identify any class imbalances.**

df=read.csv('E:/Virtual\_Intern/Dataset .csv')  
print("Number of Rows:")

## [1] "Number of Rows:"

print(nrow(df))

## [1] 9551

print("Number of Columns:")

## [1] "Number of Columns:"

print(ncol(df))

## [1] 21

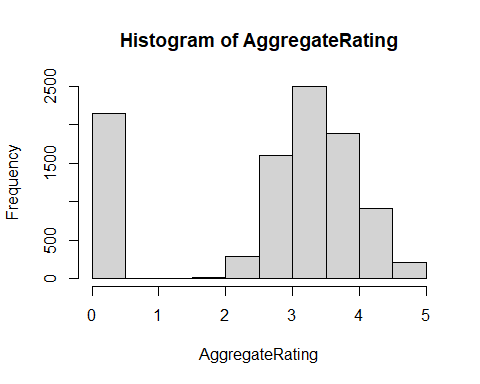
print("Missing Values:")

## [1] "Missing Values:"

print(sum(is.na(df)))

## [1] 0

AggregateRating=df$Aggregate.rating  
hist(AggregateRating)



**Task 2: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.**

print("Mean,Median and Standard Deviation Of Various Numerical Features")

## [1] "Mean,Median and Standard Deviation Of Various Numerical Features"

mean(df$Restaurant.ID,na.rm=TRUE)

## [1] 9051128

median(df$Restaurant.ID,na.rm=TRUE)

## [1] 6004089

sd(df$Restaurant.ID,na.rm=TRUE)

## [1] 8791521

mean(df$Country.Code,na.rm=TRUE)

## [1] 18.36562

median(df$Country.Code,na.rm=TRUE)

## [1] 1

sd(df$Country.Code,na.rm=TRUE)

## [1] 56.75055

mean(df$Longitude,na.rm=TRUE)

## [1] 64.12657

median(df$Longitude,na.rm=TRUE)

## [1] 77.19196

sd(df$Longitude,na.rm=TRUE)

## [1] 41.46706

mean(df$Latitude,na.rm=TRUE)

## [1] 25.85438

median(df$Latitude,na.rm=TRUE)

## [1] 28.57047

sd(df$Latitude,na.rm=TRUE)

## [1] 11.00794

mean(df$Average.Cost.for.two,na.rm=TRUE)

## [1] 1199.211

median(df$Average.Cost.for.two,na.rm=TRUE)

## [1] 400

sd(df$Average.Cost.for.two,na.rm=TRUE)

## [1] 16121.18

mean(df$Price.range,na.rm=TRUE)

## [1] 1.804837

median(df$Price.range,na.rm=TRUE)

## [1] 2

sd(df$Price.range,na.rm=TRUE)

## [1] 0.9056088

mean(df$Aggregate.rating,na.rm=TRUE)

## [1] 2.66637

median(df$Aggregate.rating,na.rm=TRUE)

## [1] 3.2

sd(df$Aggregate.rating,na.rm=TRUE)

## [1] 1.516378

mean(df$Votes,na.rm=TRUE)

## [1] 156.9097

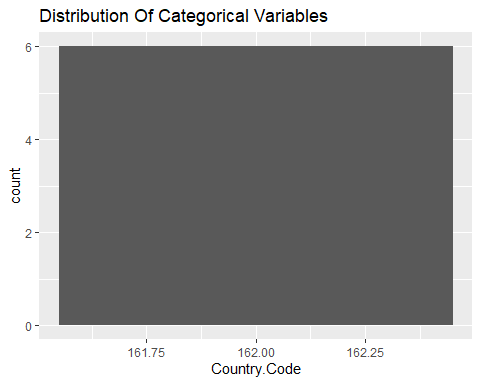
median(df$Votes,na.rm=TRUE)

## [1] 31

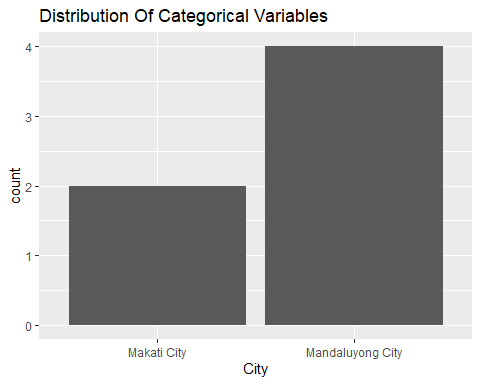
sd(df$Votes,na.rm=TRUE)

## [1] 430.1691

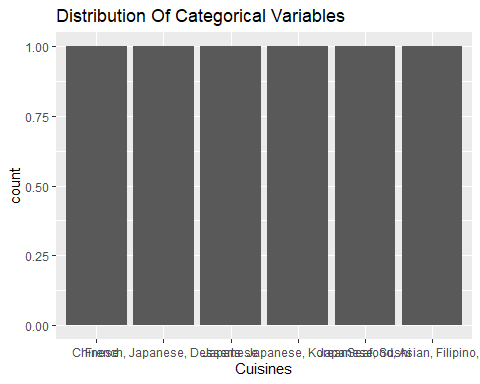
library(ggplot2)  
x1<-head(df)  
ggplot()+  
geom\_bar(data=x1,mapping = aes(x=Country.Code))+labs(title = "Distribution Of Categorical Variables")



ggplot()+  
geom\_bar(data=x1,mapping = aes(x=City))+labs(title = "Distribution Of Categorical Variables")



ggplot()+  
geom\_bar(data=x1,mapping = aes(x=Cuisines))+labs(title = "Distribution Of Categorical Variables")



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

ans<-df %>% group\_by(City) %>% summarize(count=n())  
print("Highest Number Of Restaurants with Top Cities")

## [1] "Highest Number Of Restaurants with Top Cities"

head(arrange(ans,desc(count)),10)

## # A tibble: 10 × 2  
## City count  
## <chr> <int>  
## 1 New Delhi 5473  
## 2 Gurgaon 1118  
## 3 Noida 1080  
## 4 Faridabad 251  
## 5 Ghaziabad 25  
## 6 Ahmedabad 21  
## 7 Amritsar 21  
## 8 Bhubaneshwar 21  
## 9 Guwahati 21  
## 10 Lucknow 21

ans1<-df %>% group\_by(Cuisines) %>% summarize(count=n())  
print("Highest Number Of Restaurants with Top Cuisines")

## [1] "Highest Number Of Restaurants with Top Cuisines"

head(arrange(ans1,desc(count)),10)

## # A tibble: 10 × 2  
## Cuisines count  
## <chr> <int>  
## 1 North Indian 936  
## 2 North Indian, Chinese 511  
## 3 Chinese 354  
## 4 Fast Food 354  
## 5 North Indian, Mughlai 334  
## 6 Cafe 299  
## 7 Bakery 218  
## 8 North Indian, Mughlai, Chinese 197  
## 9 Bakery, Desserts 170  
## 10 Street Food 149

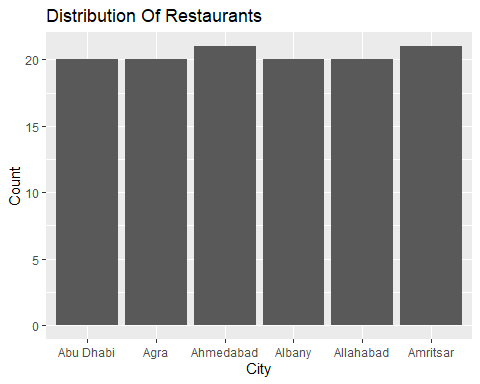
**Task 3: Geospatial Analysis**

**Visualize the locations of restaurants on a map using latitude and longitude information. Analyze the distribution of restaurants across different cities or countries. Determine if there is any correlation between the restaurant’s location and its rating.**

library(leaflet)  
map<-leaflet(df) %>% addTiles() %>% setView(lng = mean(df$Longitude),lat = mean(df$Latitude),zoom=4)  
map<-map %>% addCircleMarkers(lng = ~Longitude,lat = ~Latitude,popup=~paste("Locality:",`Locality`),radius = 3,color = 'red',stroke = FALSE,fillOpacity = 0.6)  
library(htmlwidgets)  
  
saveWidget(map,'restaurant.html',selfcontained = TRUE)  
print("Map is Saved")

## [1] "Map is Saved"

group\_Restaurant<- df %>% group\_by(Restaurant.ID)  
Group\_City<- group\_Restaurant %>% group\_by(City) %>% summarize(Count=n())  
top\_Restaurant<-head(Group\_City)  
  
ggplot(data = top\_Restaurant)+geom\_bar(mapping=aes(x=City,y=Count),stat = "identity")+labs(title = "Distribution Of Restaurants")



category\_to\_numeric<-as.numeric(factor(df$Locality))  
cor\_val<-cor(category\_to\_numeric,df$Aggregate.rating)  
if(cor\_val<0){  
 print("Datas(Locality and Aggregate Rating) are -vely Correlated")  
}else{  
 print("Datas(Locality and Aggregate Rating) are Positively Correlated")  
}

## [1] "Datas(Locality and Aggregate Rating) are -vely Correlated"

**Level 2:**

**Task 1: Table Booking and Online Delivery**

**Determine the percentage of restaurants that offer table booking and online delivery. Compare the average ratings of restaurants with table booking and those without. Analyze the availability of online delivery among restaurants with different price ranges.**

library(base)  
t1<-df$Has.Online.delivery  
Online\_percent<-prop.table(table(t1))  
Table\_booking<-prop.table(table(df$Has.Table.booking))  
print("Online Delivery")

## [1] "Online Delivery"

a1<-Online\_percent['Yes']\*100  
a1

## Yes   
## 25.66223

print("Table Booking")

## [1] "Table Booking"

a2<-Table\_booking['Yes']\*100  
a2

## Yes   
## 12.12438

avg\_rating<-aggregate(Aggregate.rating~Has.Table.booking,data=df,FUN = mean)  
print("Average Rating Of Restaurants")

## [1] "Average Rating Of Restaurants"

avg\_rating

## Has.Table.booking Aggregate.rating  
## 1 No 2.559359  
## 2 Yes 3.441969

Online\_delivery\_availability<-aggregate(Has.Online.delivery~Price.range,data=df,FUN=function(x) mean(x=='Yes')\*100)  
  
print("Availability Of Online Delivery with Different Price Ranges")

## [1] "Availability Of Online Delivery with Different Price Ranges"

Online\_delivery\_availability

## Price.range Has.Online.delivery  
## 1 1 15.774077  
## 2 2 41.310633  
## 3 3 29.190341  
## 4 4 9.044369

**Task 2: Price Range Analysis**

**Determine the most common price range among all the restaurants. Calculate the average rating for each price range. Identify the color that represents the highest average rating among different price ranges.**

tab<-table(df$Price.range)  
print("Most Common Price Range")

## [1] "Most Common Price Range"

names(tab[which.max(tab)])

## [1] "1"

print("Average Rating for each Price Range")

## [1] "Average Rating for each Price Range"

library(dplyr)  
avg\_rating\_diff\_price\_range<-df %>% group\_by(price\_range=df$Price.range) %>% summarize( Average\_Rating=mean(Aggregate.rating))  
  
avg\_rating\_diff\_price\_range

## # A tibble: 4 × 2  
## price\_range Average\_Rating  
## <int> <dbl>  
## 1 1 2.00  
## 2 2 2.94  
## 3 3 3.68  
## 4 4 3.82

highest\_avg\_rating<-avg\_rating\_diff\_price\_range %>% filter(Average\_Rating==max(Average\_Rating))  
  
color\_highest\_price\_avg\_rate<-df %>% group\_by(Rating.color) %>% filter(Price.range==highest\_avg\_rating$price\_range) %>% summarise(count=n())  
print("Color that represents the highest  
average rating among different price ranges")

## [1] "Color that represents the highest\naverage rating among different price ranges"

color\_highest\_price\_avg\_rate

## # A tibble: 6 × 2  
## Rating.color count  
## <chr> <int>  
## 1 Dark Green 74  
## 2 Green 194  
## 3 Orange 101  
## 4 Red 6  
## 5 White 11  
## 6 Yellow 200

**Task 3: Feature Engineering**

**Extract additional features from the existing columns, such as the length of the restaurant name or address. Create new features like “Has Table Booking” or “Has Online Delivery” by encoding categorical variables.**

df['Length\_of\_Restaurant\_name']<-nchar(df$Restaurant.Name)  
df['Length\_of\_Restaurant\_Address']<-nchar(df$Address)  
print("Length of Restaurant Address")

## [1] "Length of Restaurant Address"

head(df$Length\_of\_Restaurant\_Address)

## [1] 71 67 56 70 64 71

print("Length of Restaurant Name")

## [1] "Length of Restaurant Name"

head(df$Length\_of\_Restaurant\_name)

## [1] 16 16 22 4 11 12

df['Encode\_Has\_Table\_Booking']=as.numeric(factor(df$Has.Table.booking))  
print("Encoded Restaurant\_Has\_Table\_Booking")

## [1] "Encoded Restaurant\_Has\_Table\_Booking"

head(df$Encode\_Has\_Table\_Booking)

## [1] 2 2 2 1 2 1

df['Encode\_Has\_Online\_Delivery']=as.numeric(factor(df$Has.Online.delivery))  
print("Encoded Restaurant\_Has\_Online\_Delivery")

## [1] "Encoded Restaurant\_Has\_Online\_Delivery"

head(df$Encode\_Has\_Online\_Delivery)

## [1] 1 1 1 1 1 1

**Level 3:**

**Task 1: Predictive Modeling**

**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.**

train\_index<-sample(1:120,0.7\*120)  
x\_train<-df$Encode\_Has\_Table\_Booking[train\_index]  
y\_train<-df$Aggregate.rating[train\_index]  
x\_test<-df$Encode\_Has\_Table\_Booking[-train\_index]  
y\_test<-df$Aggregate.rating[-train\_index]  
df\_train<-data.frame(x=x\_train,y=y\_train)  
df\_test<-data.frame(x=x\_test,y=y\_test)  
lm\_model<-function(df\_train){  
 beta1<-sum((df\_train$x-mean(df\_train$x))\*(df\_train$y-mean(df\_train$y)))/sum((df\_train$x-mean(df\_train$x))^2)  
 beta0<-mean(df\_train$y)-beta1\*mean(df\_train$x)  
 return(c(x1=beta0,y1=beta1))  
}  
ans<-lm\_model(df\_train)  
print("Slope and Intercept From Linear Regression Model")

## [1] "Slope and Intercept From Linear Regression Model"

ans

## x1 y1   
## 3.1110811 0.6794595

lr\_predict<-function(ans,df\_test)  
{  
 y\_pred<-ans["x1"]+ans["y1"]\*df\_test$x  
 return(data.frame(pred=y\_pred))  
}  
ans1<-lr\_predict(ans,df\_test)  
  
print("Prediction Of Aggregate Rating based On Features")

## [1] "Prediction Of Aggregate Rating based On Features"

glimpse(head(ans1))

## Rows: 6  
## Columns: 1  
## $ pred <dbl> 3.790541, 4.470000, 3.790541, 4.470000, 3.790541, 4.470000

#Performance Evaluation  
mse<-mean((df\_test$y-ans1$pred)^2)  
mse<-sqrt(mse)  
  
print("The Evaluated Performance Metrices based on Root Mean Square Error")

## [1] "The Evaluated Performance Metrices based on Root Mean Square Error"

mse

## [1] 1.923761

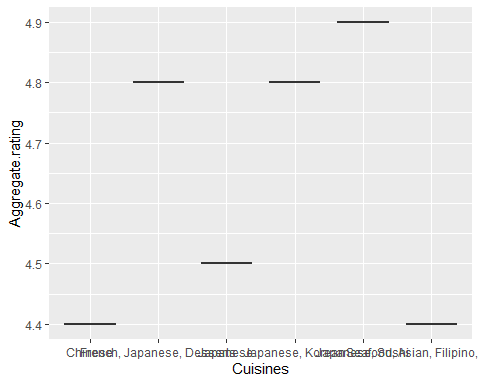
**Task 2: Customer Preference**

**Analysis Analyze the relationship between the type of cuisine and the restaurant’s rating. Identify the most popular cuisines among customers based on the number of votes. Determine if there are any specific cuisines that tend to receive higher ratings.**

ans<-head(df)  
ggplot(data =ans)+geom\_bar(mapping = aes(x=Cuisines,y=Aggregate.rating),stat = "identity")



ggplot(data =ans)+geom\_boxplot(mapping = aes(x=Cuisines,y=Aggregate.rating))



print("Most Popular Cuisines")

## [1] "Most Popular Cuisines"

df %>% filter(Votes==max(Votes)) %>% summarise(Cuisines)

## Cuisines  
## 1 Italian, American, Pizza

print("Specific Cuisines to receive high Ratings")

## [1] "Specific Cuisines to receive high Ratings"

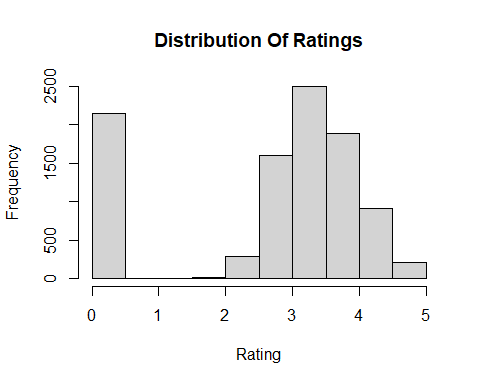
df %>% filter(Aggregate.rating==max(Aggregate.rating)) %>% reframe(Cuisines)

## Cuisines  
## 1 Japanese, Sushi  
## 2 European, Asian, Indian  
## 3 Filipino, Mexican  
## 4 International  
## 5 Brazilian, Bar Food  
## 6 Brazilian, Bar Food  
## 7 American, Caribbean, Seafood  
## 8 Burger  
## 9 BBQ, Breakfast, Southern  
## 10 Asian  
## 11 American, Coffee and Tea  
## 12 Sandwich, Seafood, Cajun  
## 13 Pizza, Sandwich  
## 14 American, Sandwich, Tea  
## 15 American, BBQ, Sandwich  
## 16 Burger, Bar Food, Steak  
## 17 Hawaiian, Seafood  
## 18 Japanese  
## 19 Italian, Deli  
## 20 European, German  
## 21 Indian, North Indian  
## 22 Continental, Indian  
## 23 Indian  
## 24 Indian  
## 25 Cafe, North Indian, Chinese  
## 26 Fast Food  
## 27 North Indian, European, Mediterranean  
## 28 Bakery, Desserts  
## 29 North Indian  
## 30 Mexican, American, Healthy Food  
## 31 North Indian  
## 32 European, Mediterranean, North Indian  
## 33 European, Mediterranean, North Indian  
## 34 Italian, Bakery, Continental  
## 35 North Indian, Chinese  
## 36 North Indian, Chinese  
## 37 Mughlai, Lucknowi  
## 38 North Indian, South Indian, Mughlai  
## 39 North Indian, European, Mediterranean  
## 40 Ice Cream  
## 41 Modern Indian  
## 42 Modern Indian  
## 43 North Indian, Chinese, Mediterranean  
## 44 Sunda, Indonesian  
## 45 Sushi, Japanese  
## 46 Sunda, Indonesian  
## 47 Sunda, Indonesian  
## 48 Desserts  
## 49 Desserts  
## 50 Steak  
## 51 British  
## 52 Taiwanese, Street Food  
## 53 American, Burger, Grill  
## 54 Chinese  
## 55 European, Contemporary  
## 56 Tapas  
## 57 French  
## 58 Seafood  
## 59 World Cuisine  
## 60 Cafe  
## 61 Bar Food

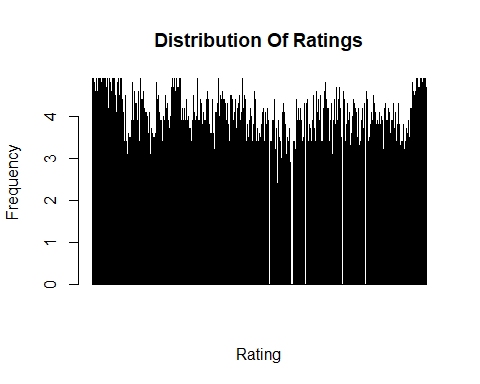
**Task 3: Data Visualization**

**Create visualizations to represent the distribution of ratings using different charts (histogram, bar plot, etc.). Compare the average ratings of different cuisines or cities using appropriate visualizations. Visualize the relationship between various features and the target variable to gain insights.**

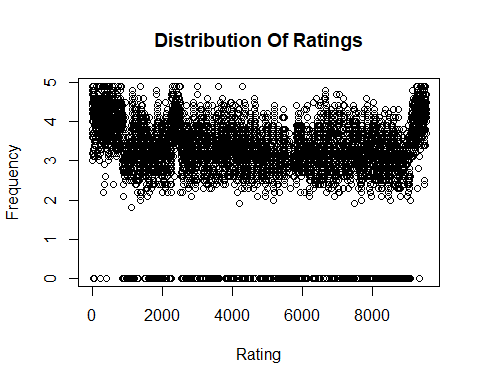
hist(df$Aggregate.rating,main = "Distribution Of Ratings",xlab = "Rating",ylab = "Frequency")



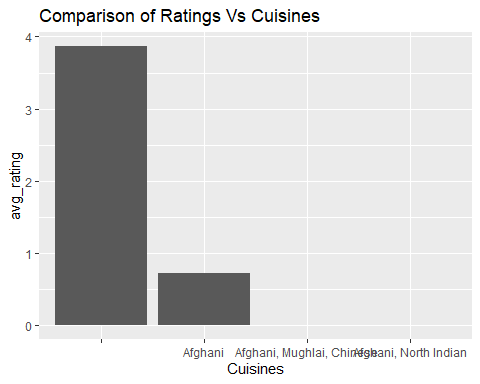
barplot(df$Aggregate.rating,main = "Distribution Of Ratings",xlab = "Rating",ylab = "Frequency")



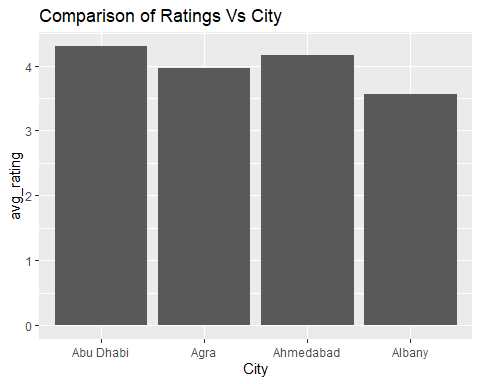
plot(df$Aggregate.rating,main = "Distribution Of Ratings",xlab = "Rating",ylab = "Frequency")



avg\_rate\_diff\_cuisine<-head(df %>% group\_by(Cuisines) %>% summarise(avg\_rating=mean(Aggregate.rating)),4)  
  
library(ggplot2)  
ggplot(data=avg\_rate\_diff\_cuisine,mapping=aes(x=Cuisines,y=avg\_rating))+geom\_bar(stat="identity")+labs(title = "Comparison of Ratings Vs Cuisines")



avg\_rate\_diff\_city<-head(df %>% group\_by(City) %>% summarise(avg\_rating=mean(Aggregate.rating)),4)  
  
ggplot(data=avg\_rate\_diff\_city,mapping=aes(x=City,y=avg\_rating))+geom\_bar(stat="identity")+labs(title = "Comparison of Ratings Vs City")



ggplot(data=df)+geom\_boxplot(mapping=aes(x=Cuisines,y=Aggregate.rating))+coord\_cartesian(xlim = c(0,3))

