install.packages("tidyr")

install.packages("tidyverse")

install.packages("dplyr")

install.packages("reshape2")

install.packages("randomForest")

install.packages("caret")

library(tidyr)

library(dplyr)

library(magrittr)

library(ggplot2)

library(reshape2)

library(randomForest)

library(caret)

#[1]

#Importing the data set

raw\_data <- read.csv("C:/Users/Sarthak Gupta/Desktop/RProg/archive/hotel\_booking.csv")

print(raw\_data)

nrow(raw\_data)

#Inspecting the data set for structure

str(raw\_data)

summary(raw\_data)

head(raw\_data)

#[2]

#Check for missing values

colSums(is.na(raw\_data))

#Removing rows with missing values

hotel\_data <- raw\_data %>%

mutate(across(where(is.numeric), ~ replace\_na(., 0))) %>%

na.omit()

nrow(hotel\_data)

colSums(is.na(hotel\_data))

#Converting data types

hotel\_data <- hotel\_data %>%

mutate\_at(vars(hotel, arrival\_date\_month, meal, country, market\_segment,

distribution\_channel, reserved\_room\_type, assigned\_room\_type,

deposit\_type, customer\_type, reservation\_status), as.factor) %>%

mutate\_at(vars(children, agent, company), as.integer) %>%

mutate(reservation\_status\_date = as.Date(reservation\_status\_date))

str(hotel\_data)

#Remove irrelevant columns

hotel\_data <- hotel\_data %>%

select(-c(company, phone.number, name, email, credit\_card, deposit\_type,

distribution\_channel, babies, previous\_bookings\_not\_canceled))

str(hotel\_data)

nrow(hotel\_data)

#[3]

#EDA

#Descriptive statistics

numeric\_data <- hotel\_data %>%

select(where(is.numeric))

descriptive\_stats <- numeric\_data %>%

summarise(across(everything(),

list(Mean = mean, Median = median, SD = sd), na.rm = TRUE))

print(descriptive\_stats)

# Bar plot for arrival months

hotel\_data$arrival\_date\_month <- factor(hotel\_data$arrival\_date\_month,

levels = c("January", "February", "March", "April",

"May", "June", "July", "August",

"September", "October", "November", "December"))

ggplot(hotel\_data, aes(x = arrival\_date\_month)) +

geom\_bar(fill = "steelblue") +

labs(title = "Count of Arrivals by Month",

x = "Arrival Month",

y = "Count") +

theme\_minimal()

# Correlation matrix

corr\_matrix <- cor(numeric\_data, method = "pearson")

corr\_melted <- melt(corr\_matrix)

ggplot(corr\_melted, aes(Var1, Var2, fill = value)) +

geom\_tile(color = "black") + # Create the heatmap tiles with borders

geom\_text(aes(label = sprintf("%.2f", value)), color = "black", size = 3) + # Display correlation values

scale\_fill\_gradient2(low = "white", high = "red", mid = "lightblue",

midpoint = 0, limit = c(-1, 1),

name = "Correlation") +

labs(title = "Correlation Matrix", x = "", y = "") +

theme\_minimal() +

theme(axis.text.x = element\_text(angle = 45, hjust = 1)) # Rotate x-axis labels for better readability

#[4]

##Taking is\_cancelled as target variable

hotel\_data$is\_canceled <- as.factor(hotel\_data$is\_canceled)

# Using 70-30 rule for training and testing

set.seed(123)

train\_index <- createDataPartition(hotel\_data$is\_canceled, p = 0.7, list = FALSE)

train\_data <- hotel\_data[train\_index, ]

test\_data <- hotel\_data[-train\_index, ]

# Random Forest model

rf\_model <- randomForest(is\_canceled ~ lead\_time + adr + stays\_in\_weekend\_nights + total\_of\_special\_requests +

stays\_in\_week\_nights + adults,

data = train\_data,

ntree = 100, # Number of trees

mtry = 2, # Number of variables to consider at each split

importance = TRUE) # To get variable importance

#[5]

#Model Evaluation

# Make predictions on the testing set

predictions <- predict(rf\_model, newdata = test\_data)

confusion\_matrix <- confusionMatrix(predictions, test\_data$is\_canceled)

print(confusion\_matrix)

accuracy <- sum(predictions == test\_data$is\_canceled) / nrow(test\_data)

cat("Accuracy: ", accuracy, "\n")

#Importance of features##

# Get the importance of the features

importance\_rf <- importance(rf\_model)

print(importance\_rf)

# Visualize the importance

varImpPlot(rf\_model, main = "Variable Importance in Random Forest")

# Create a confusion matrix

confusion\_matrix <- confusionMatrix(predictions, test\_data$is\_canceled)

print(confusion\_matrix)

# Visualize the confusion matrix

conf\_matrix\_table <- as.data.frame(confusion\_matrix$table)

ggplot(conf\_matrix\_table, aes(x = Reference, y = Prediction, fill = Freq)) +

geom\_tile(color = "black") +

geom\_text(aes(label = Freq), color = "white") + # Add the counts inside the tiles

scale\_fill\_gradient(low = "lightblue", high = "salmon") +

labs(title = "Confusion Matrix",

x = "Actual",

y = "Predicted") +

theme\_minimal()

#Graphs:

# Bar plot for hotel cancellations

ggplot(hotel\_data, aes(x = hotel, fill = as.factor(is\_canceled))) +

geom\_bar(position = "dodge", na.rm = TRUE) +

scale\_fill\_manual(values = c("darkgreen", "orange"),

labels = c("Not Canceled", "Canceled")) +

labs(title = "Hotel Cancellations",

x = "Hotel",

y = "Count",

fill = "Cancellation Status")+

theme\_minimal() +

theme(legend.position = "top")

custom\_palette <- c("Resort Hotel" = "lightblue", "City Hotel" = "salmon")

#Box plot for ADR by hotel type

ggplot(hotel\_data, aes(x = hotel, y = adr, fill = hotel)) +

geom\_boxplot() +

labs(title = "Average Daily Rate (ADR) by Hotel",

x = "Hotel",

y = "Average Daily Rate (ADR)") +

theme\_minimal()

#Scatter plot for lead time vs cancellations

ggplot(hotel\_data, aes(x = lead\_time,

y = factor(is\_canceled, levels = c(0, 1), labels = c("Not Canceled", "Canceled")),

color = factor(is\_canceled, levels = c(0, 1), labels = c("Not Canceled", "Canceled")))) +

geom\_jitter(alpha = 0.5) +

labs(title = "Lead Time vs. Booking Cancellations",

x = "Lead Time (days)",

y = "Cancellation Status") +

scale\_color\_manual(values = c("Not Canceled" = "lightblue", "Canceled" = "salmon")) +

theme\_minimal() +

theme(legend.position = "top")