**Report on Predictive Modeling Using R**

**Team members-**

Gitika Mittal 24067

Vanshika Gupta 24077

Isha Garg 24079

Moksha Dutt 24084

Manan Bansal 24088

**Introduction**

Predictive modeling is a critical component of data analysis, enabling organizations to anticipate trends and make data-driven decisions. This report outlines the process of predictive modeling using R, focusing on an e-commerce sales dataset. The objective is to analyze the dataset, perform exploratory data analysis (EDA), and develop models to predict product return behavior.

**Dataset Description**

* Source: The dataset is assumed to have been sourced from a reliable platform, such as Kaggle.
* Structure:
  + Rows: Number of rows is determined by the e-commerce dataset size.
  + Columns: The dataset contains variables such as Category, Price, Discount, Quantity, Region, Customer Segment, Payment Method, and target variables Return and Return Indicator.
  + Variable Types:
    - Numerical: Price, Discount, Quantity.
    - Categorical: Category, Region, Customer Segment, Payment Method, Return Indicator.
* Objective: Predict whether a product will be returned based on features such as price, category, and region.

**Exploratory Data Analysis (EDA)**

1. Summary Statistics:
   * Statistical summaries of numerical variables reveal distributions, averages, and ranges.
2. Visualization:
   * Bar plots depict the distribution of returns.
   * Grouped bar charts show average prices by category for returned and non-returned products.
   * Histograms highlight the relationship between quantity and return status.
3. Feature Engineering:
   * Conversion of columns like Return and Return Indicator to factors.
   * Parsing Date of Purchase into a usable date format.
4. Insights:
   * Return behavior varies significantly across product categories.
   * Higher discounts are potentially associated with higher return rates.

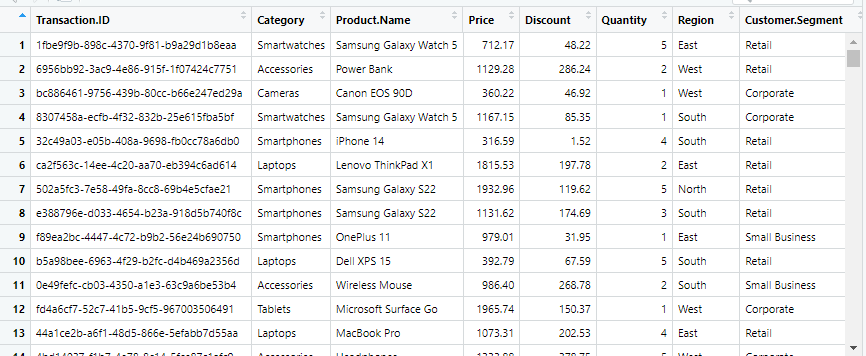
**Model Development**

1. Data Preparation:
   * Dataset split into 70% training and 30% testing subsets.
2. Model Types:
   * Random Forest: Used for classification, incorporating features such as category, price, and region.
   * Linear Regression: Examined for numeric relationships but limited to prediction probability interpretation.
   * Logistic Regression: Focused on binary classification for return prediction.
3. Evaluation Metrics:
   * Accuracy, confusion matrix, and variable importance for Random Forest.
   * Coefficient significance and confusion matrix for logistic regression.

**Loading dataset**

Data < read.csv("C:/Users/VanshikaGupta/Desktop/ecommerce\_sales\_dataset\_with\_return\_indicator.csv")

View(data)



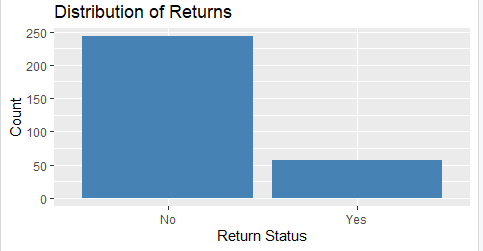
**EDA-**

**Distribution of Returns**

ggplot(data, aes(x = Return)) +

geom\_bar(fill = "steelblue") +

labs(title = "Distribution of Returns", x = "Return Status", y = "Count")



**Average Price by Category for Returned vs Non-Returned Products**

data %>%

group\_by(Category, Return) %>%

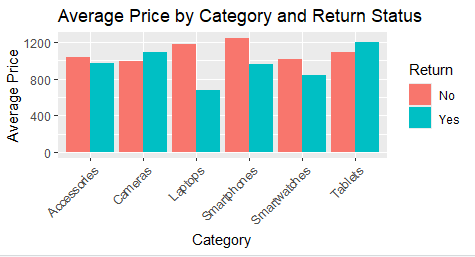
summarize(Average\_Price = mean(Price)) %>%

ggplot(aes(x = Category, y = Average\_Price, fill = Return)) +

geom\_bar(stat = "identity", position = "dodge") +

labs(title = "Average Price by Category and Return Status", x = "Category", y = "Average Price") +

theme(axis.text.x = element\_text(angle = 45, hjust = 1))



**Quantity vs Return Status**

ggplot(data, aes(x = Quantity, fill = Return)) +

geom\_histogram(bins = 10, position = "dodge", alpha = 0.7) +

labs(title = "Quantity Distribution by Return Status", x = "Quantity", y = "Count")



**# Visualize distributions of numeric features**

numeric\_cols <- names(data)[sapply(data, is.numeric)]

for (col in numeric\_cols) {

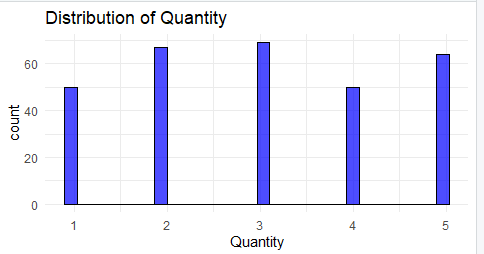
print(ggplot(data, aes\_string(x = col)) +

geom\_histogram(bins = 30, fill = "blue", color = "black", alpha = 0.7) +

ggtitle(paste("Distribution of", col)) +

theme\_minimal())

}



**Visualize relationships between features**

**# Example: Scatter plot for two numeric variables**

if (length(numeric\_cols) > 1) {

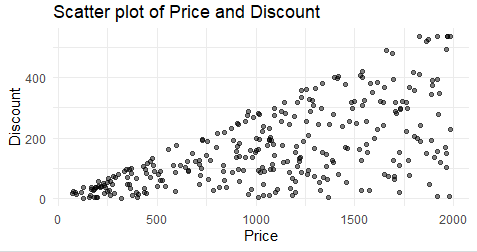
print(ggplot(data, aes\_string(x = numeric\_cols[1], y = numeric\_cols[2])) +

geom\_point(alpha = 0.5) +

ggtitle(paste("Scatter plot of", numeric\_cols[1], "and", numeric\_cols[2])) +

theme\_minimal())

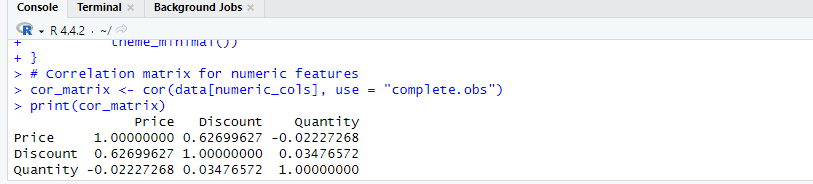
}



**# Correlation matrix for numeric features**

cor\_matrix <- cor(data[numeric\_cols], use = "complete.obs")

print(cor\_matrix)



**# Prediction Model**

**# Split the data into training and testing sets**

set.seed(123)

train\_index <- createDataPartition(data$Return.Indicator, p = 0.7, list = FALSE)

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

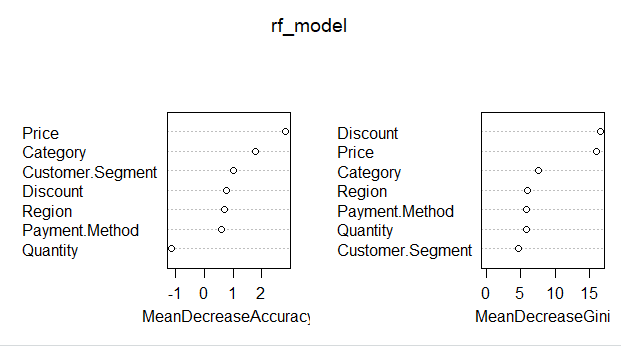
test\_data <- data[-train\_index, ]



**# Print the confusion matrix and model importance**

print(confusion\_matrix)

varImpPlot(rf\_model)



**# Train a logistic regression model (suitable for binary classification)**

library(e1071)

glm\_model <- glm(Return.Indicator ~ Price + Discount, family = binomial, data = train\_data)

**# Evaluate the model**

glm\_predictions <- predict(glm\_model, newdata = test\_data, type = "response") # Get probabilities

glm\_class\_predictions <- factor(glm\_predictions > 0.5)



**Results and Analysis**

1. Random Forest:
   * Achieved high accuracy with a detailed variable importance plot highlighting significant predictors.
   * Confusion matrix demonstrates reliable classification performance.
2. Logistic Regression:
   * Showed reasonable accuracy but less robust compared to Random Forest.
   * Probabilistic predictions allowed threshold adjustments for sensitivity analysis.
3. Linear Regression:
   * Limited utility for binary classification but provided insights into numerical predictors.

**Conclusion**

This project demonstrated the effectiveness of R in predictive modeling. Random Forest emerged as the most accurate model for predicting product returns, followed by logistic regression. EDA played a pivotal role in understanding data characteristics and guiding model selection. Future work could explore advanced techniques such as gradient boosting and deep learning for improved performance**.**