FA 6

Lindsay Faith Bazar

May 04, 2025

Data Exploration

```
glimpse(data)
```

```
## Rows: 10,532
## Columns: 8
## $ `Customer ID`                        <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,~
## $ Age                                  <dbl> 56, 69, 46, 32, 60, 25, 38, 56, 36~
## $ `Annual Income (K$)`                 <dbl> 106, 66, 110, 50, 73, 48, 100, 131~
## $ Gender                               <chr> "Female", "Female", "Male", "Male"~
## $ `Product Category Purchased`         <chr> "Fashion", "Home", "Fashion", "Ele~
## $ `Average Spend per Visit ($)`        <dbl> 163.45276, 163.02050, 104.54128, 1~
## $ `Number of Visits in Last 6 Months` <dbl> 16, 31, 29, 26, 38, 22, 20, 33, 34~
## $ `Customer Segment`                   <chr> "Premium Shopper", "Budget Shopper~
```

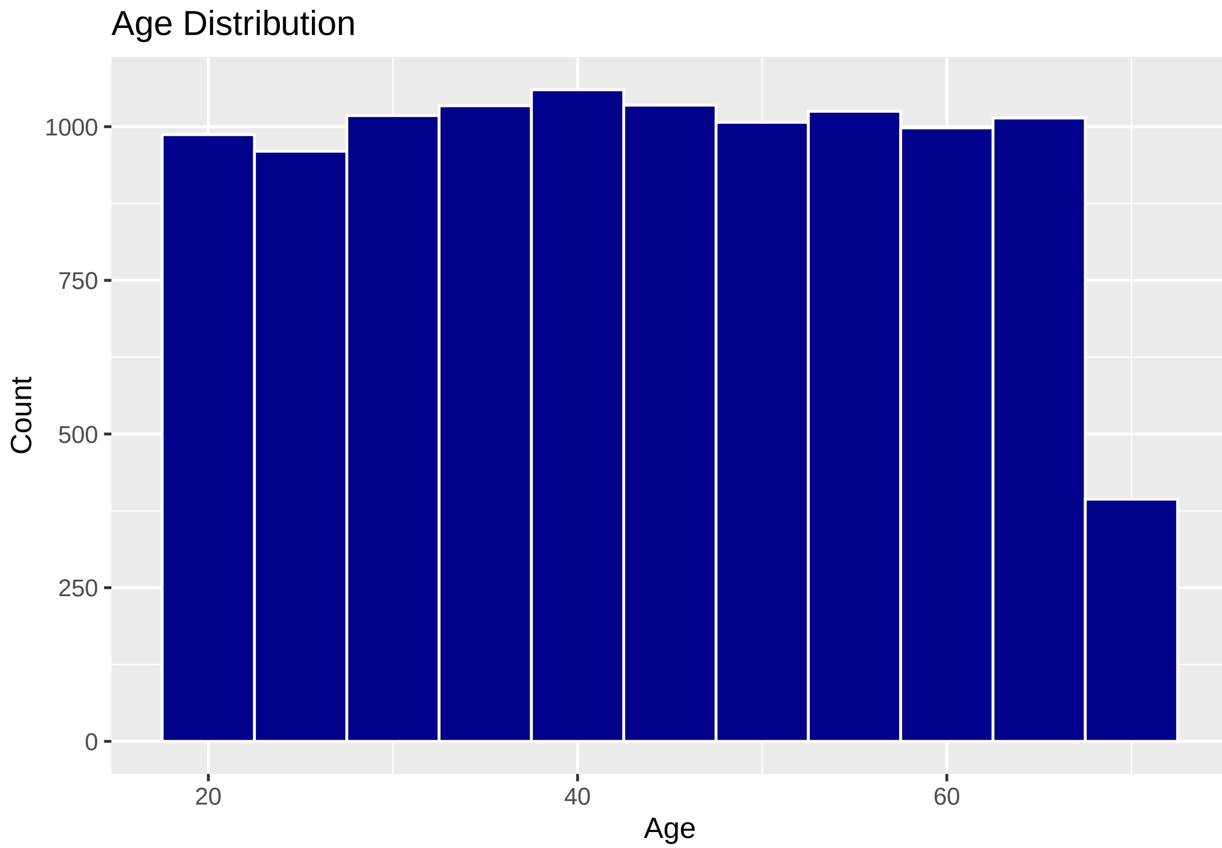
```
colSums(is.na(data))
```

```
##                       Customer ID                               Age
##                                 0                                 0
##                Annual Income (K$)                            Gender
##                                 0                                 0
##        Product Category Purchased       Average Spend per Visit ($)
##                                 0                                 0
## Number of Visits in Last 6 Months                  Customer Segment
##                                 0                                 0
```

Since there are no missing values present, we can proceed to the visualization step.

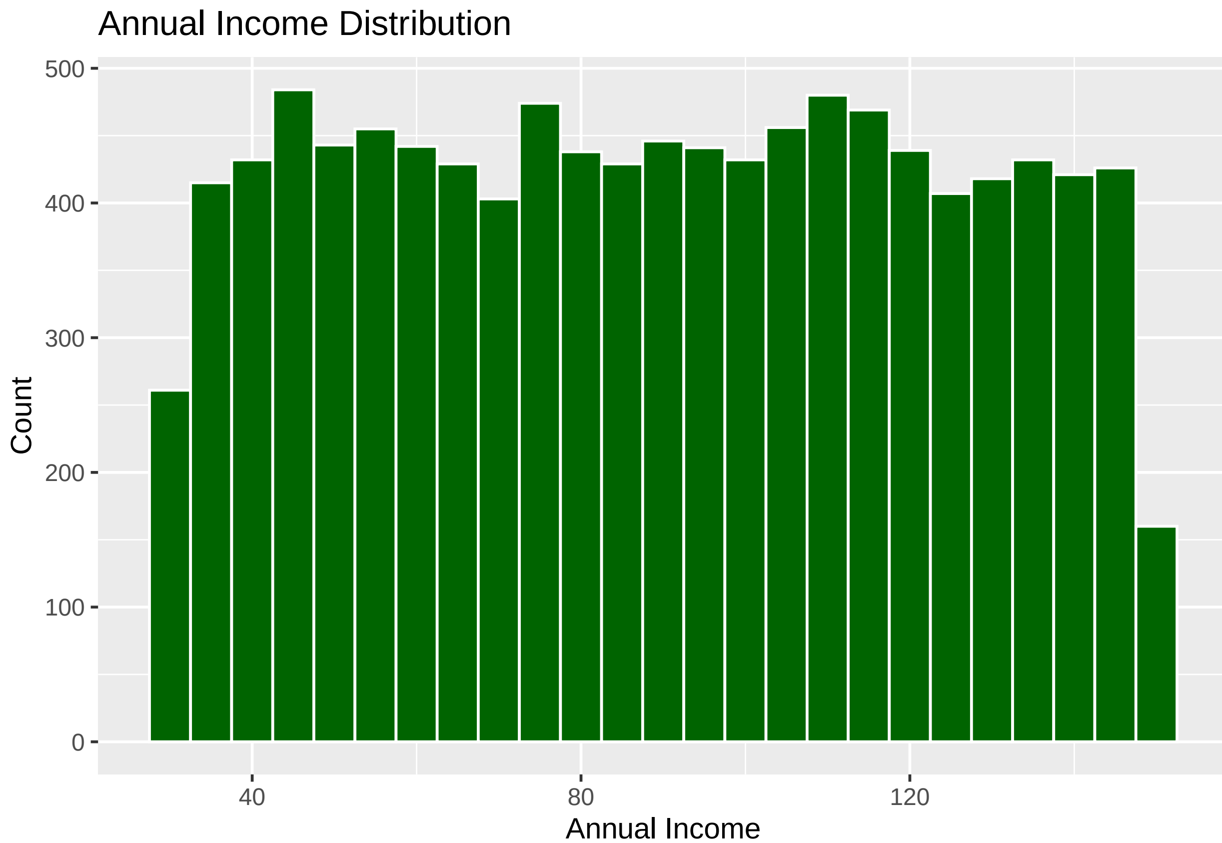
Age Distribution

```
ggplot(data, aes(x=Age)) +
  geom_histogram(binwidth = 5, fill = "darkblue", color = "white") +
  labs(title = "Age Distribution", x = "Age", y = "Count")
```



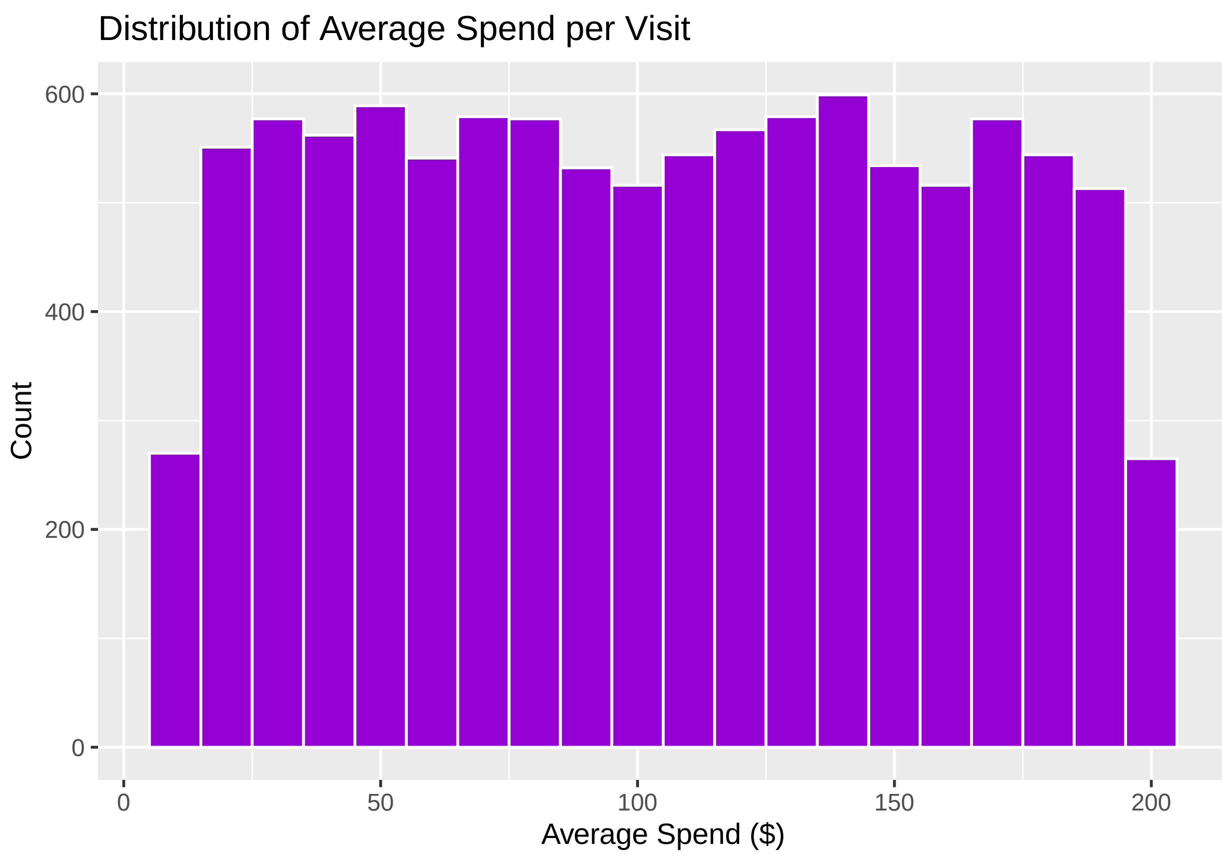
Annual Income Distribution

```
ggplot(data, aes(x=`Annual Income (K$)`)) +
  geom_histogram(binwidth = 5, fill = "darkgreen", color = "white") +
  labs(title = "Annual Income Distribution" , x = "Annual Income", y = "Count")
```

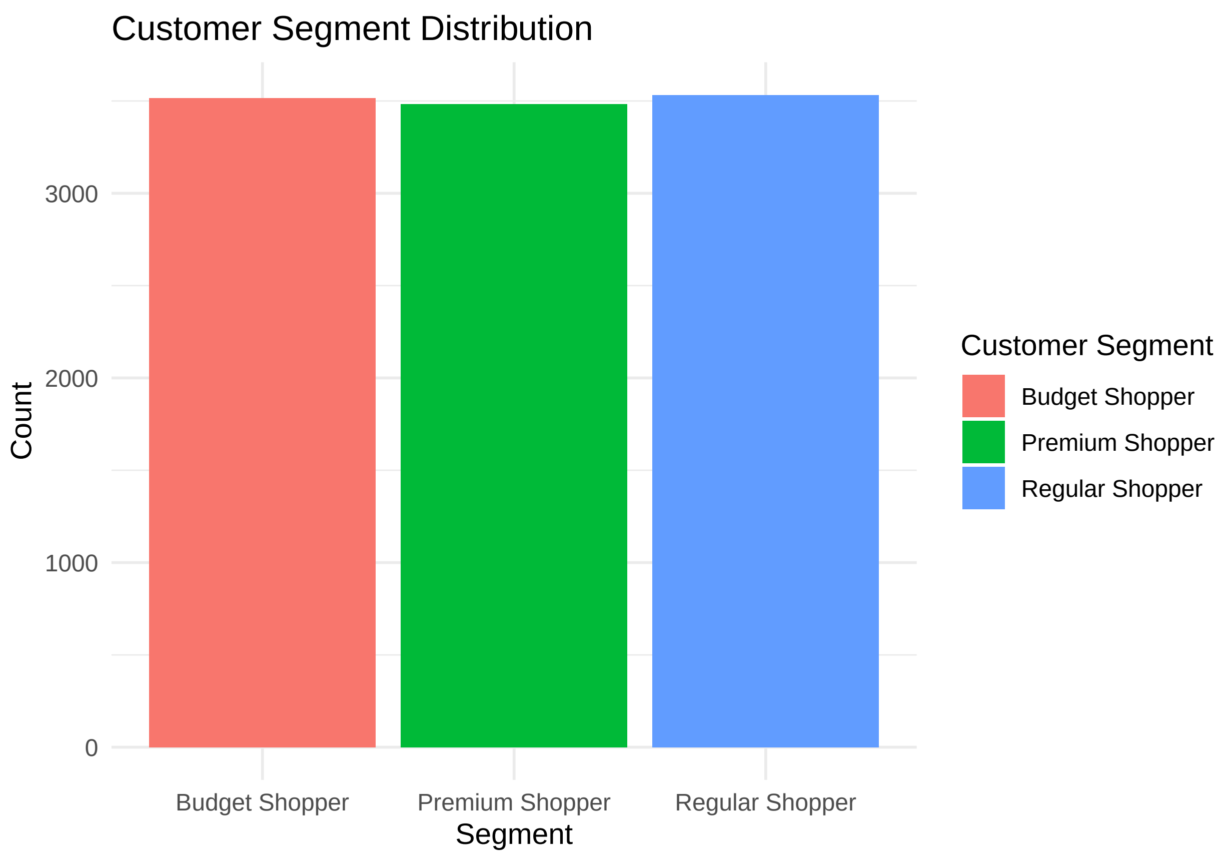


Average Spend per Visit Distribution

```r
ggplot(data, aes(x = `Average Spend per Visit ($)`)) +
  geom_histogram(binwidth = 10, fill = "darkviolet", color = "white") +
  labs(title = "Distribution of Average Spend per Visit", x = "Average Spend ($)", y = "Count")
```




```r
ggplot(data, aes(x = `Customer Segment`, fill = `Customer Segment`)) +
  geom_bar() +
  labs(title = "Customer Segment Distribution", x = "Segment", y = "Count") +
  theme_minimal()
```



Data Preprocessing

Encoding gender to numeric:

```
data$Gender <- ifelse(data$Gender == "Male", 1, 0)
```

One-Hot Encoding for the product category:

```
data <- data%>%
  mutate(`Product Category Purchased` = as.factor(`Product Category Purchased`)) %>%
  tidyr::pivot_wider(
    names_from = `Product Category Purchased`,
    values_from = `Product Category Purchased`,
    values_fn = length,
    values_fill = 0
  )
```

Scaling Numeric Variables:

```
data_scaled <- data %>%
  mutate(
    Age = scale(Age),
    `Annual Income (K$)` = scale(`Annual Income (K$)`),
    `Average Spend per Visit ($)` = scale(`Average Spend per Visit ($)`)
  )
```

Converting Target Variable to Factor:

```
data_scaled$`Customer Segment` <- as.factor(data_scaled$`Customer Segment`)
```

Splitting into Training and Test Sets:

```
set.seed(123)
train_index <- createDataPartition(data_scaled$`Customer Segment`, p = 0.8, list = FALSE)

train_data <- data_scaled[train_index, ]
test_data  <- data_scaled[-train_index, ]
```

Model Building

```
model <- multinom(`Customer Segment` ~ ., data = train_data)
```

```
## # weights:  39 (24 variable)
## initial  value 9258.005757
## iter  10 value 9253.002577
## iter  20 value 9249.273522
## final  value 9248.931418
## converged
```

```
summary(model)
```

```
## Call:
## multinom(formula = `Customer Segment` ~ ., data = train_data)
##
## Coefficients:
##                 (Intercept) `Customer ID`        Age `Annual Income (K$)`
## Premium Shopper  0.05884859 -2.724024e-06 0.01622848          -0.02799553
## Regular Shopper  0.05944282 -3.175381e-06 0.01033032          -0.03992464
##                      Gender `Average Spend per Visit ($)`
## Premium Shopper -0.03718886                   -0.01397102
## Regular Shopper -0.06187535                   -0.03680408
##                 `Number of Visits in Last 6 Months`    Fashion       Home
## Premium Shopper                       -0.0020453883 0.13159286 0.02319590
## Regular Shopper                       -0.0008337835 0.07463887 0.01128578
##                 Electronics     Others      Books
## Premium Shopper 0.001683397 0.01586643 -0.1134900
## Regular Shopper 0.019854443 0.06882157 -0.1151578
##
## Std. Errors:
##                  (Intercept) `Customer ID`        Age `Annual Income (K$)`
## Premium Shopper 0.0001801517  7.085075e-06 0.01327365           0.01332889
## Regular Shopper 0.0001792607  7.069207e-06 0.01336463           0.01341919
##                       Gender `Average Spend per Visit ($)`
## Premium Shopper 0.0001242479                    0.01331658
## Regular Shopper 0.0001169646                    0.01340279
##                 `Number of Visits in Last 6 Months`      Fashion         Home
```

```
## Premium Shopper                        0.001787296 0.0001407795 8.002041e-05
## Regular Shopper                        0.001778354 0.0001384202 8.232698e-05
##                   Electronics       Others        Books
## Premium Shopper 0.0001306963 1.713746e-05 0.0001904772
## Regular Shopper 0.0001312811 1.763691e-05 0.0001895931
##
## Residual Deviance: 18497.86
## AIC: 18541.86
```

The multinomial logistic regression model predicts customer segments based on their personal details and shopping habits. Buying fashion products makes it more likely for a customer to be a Premium or Regular shopper. Female customers are also more likely to be Premium shoppers compared to Regular ones. Older customers have a slightly higher chance of being Premium shoppers. Interestingly, higher annual income seems to lower the chances of being Premium, which suggests that income data might need to be scaled for better results.

Residual Deviance: 18, 497.86 AIC: 18,541.86

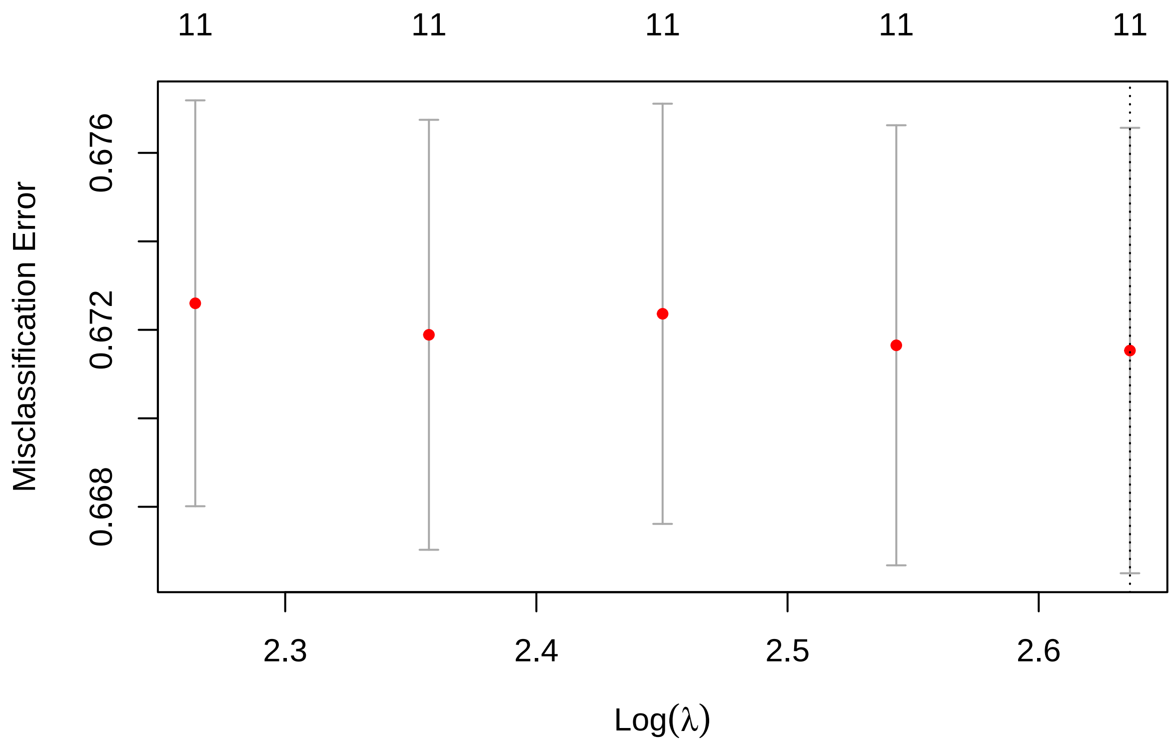
Tuning hyperparameters using cross-validation:

```
y <- as.factor(train_data$`Customer Segment`)
x <- model.matrix(`Customer Segment` ~ . - 1, data = train_data)


x_test <- model.matrix(`Customer Segment` ~ . - 1, data = test_data)
y_test <- as.factor(test_data$`Customer Segment`)

cv_model <- cv.glmnet(
  x, y,
  family = "multinomial",
  type.measure = "class",
  alpha = 0,
  nfolds = 5
)

plot(cv_model)
```




```r
best_lambda <- cv_model$lambda.min
print(best_lambda)
```

```
## [1] 13.96185
```

```r
final_model <- glmnet(
  x, y,
  family = "multinomial",
  alpha = 0,
  lambda = best_lambda
)
```

Model Evaluation

```r
predictions <- predict(final_model, newx = x_test, type = "class")
```

```r
conf_mat <- confusionMatrix(factor(predictions), y_test)
```

```
## Warning in levels(reference) != levels(data): longer object length is not a
## multiple of shorter object length

## Warning in confusionMatrix.default(factor(predictions), y_test): Levels are not
## in the same order for reference and data. Refactoring data to match.
```

```r
conf_mat
```

```
## Confusion Matrix and Statistics
##
##                  Reference
## Prediction        Budget Shopper Premium Shopper Regular Shopper
##   Budget Shopper              28              25              26
##   Premium Shopper              0               0               0
##   Regular Shopper            675             671             680
##
## Overall Statistics
##
##                Accuracy : 0.3363
##                  95% CI : (0.3162, 0.357)
##     No Information Rate : 0.3354
##     P-Value [Acc > NIR] : 0.4714
##
##                   Kappa : 0.0015
##
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: Budget Shopper Class: Premium Shopper
## Sensitivity                        0.03983                 0.0000
## Specificity                        0.96362                 1.0000
## Pos Pred Value                     0.35443                    NaN
## Neg Pred Value                     0.66683                 0.6694
## Prevalence                         0.33397                 0.3306
## Detection Rate                     0.01330                 0.0000
## Detection Prevalence               0.03753                 0.0000
## Balanced Accuracy                  0.50173                 0.5000
##                      Class: Regular Shopper
## Sensitivity                         0.96317
## Specificity                         0.03788
## Pos Pred Value                      0.33564
## Neg Pred Value                      0.67089
## Prevalence                          0.33539
## Detection Rate                      0.32304
## Detection Prevalence                0.96247
## Balanced Accuracy                   0.50053
```

```r
accuracy <- conf_mat$overall["Accuracy"]
precision <- conf_mat$byClass[, "Pos Pred Value"]
recall <- conf_mat$byClass[, "Sensitivity"]
f1_score <- 2 * (precision * recall) / (precision + recall)


cat("Accuracy:", round(accuracy, 4), "\n")
```

```
## Accuracy: 0.3363
```

```r
cat("Precision (per class):\n"); print(round(precision, 4))
```

```
## Precision (per class):

##  Class: Budget Shopper Class: Premium Shopper Class: Regular Shopper
##                 0.3544                    NaN                 0.3356
```

```r
cat("Recall (per class):\n"); print(round(recall, 4))
```

```
## Recall (per class):

##  Class: Budget Shopper Class: Premium Shopper Class: Regular Shopper
##                 0.0398                 0.0000                 0.9632
```

```r
cat("F1-Score (per class):\n"); print(round(f1_score, 4))
```

```
## F1-Score (per class):

##  Class: Budget Shopper Class: Premium Shopper Class: Regular Shopper
##                 0.0716                    NaN                 0.4978
```

Refinement

```r
data_scaled <- data_scaled %>%
  mutate(
    Income_Age_Interaction = scale(`Annual Income (K$)` * Age)
  )
set.seed(123)
train_index <- createDataPartition(data_scaled$`Customer Segment`, p = 0.8, list = FALSE)
train_data <- data_scaled[train_index, ]
test_data <- data_scaled[-train_index, ]

x_train <- model.matrix(`Customer Segment` ~ . -1, data = train_data)
y_train <- as.factor(train_data$`Customer Segment`)
x_test <- model.matrix(`Customer Segment` ~ . -1, data = test_data)
y_test <- as.factor(test_data$`Customer Segment`)

alphas <- seq(0, 1, by = 0.2)  # From Ridge (0) to LASSO (1)
cv_results <- list()

for (a in alphas) {
  cat("Fitting model with alpha =", a, "\n")
  cv_fit <- cv.glmnet(
    x_train, y_train,
    family = "multinomial",
    type.measure = "class",
    alpha = a,
    nfolds = 5
  )
  cv_results[[paste0("alpha_", a)]] <- cv_fit
}
```

```
## Fitting model with alpha = 0
## Fitting model with alpha = 0.2
## Fitting model with alpha = 0.4
## Fitting model with alpha = 0.6
## Fitting model with alpha = 0.8
## Fitting model with alpha = 1
```

```
best_model <- NULL
lowest_error <- Inf
best_alpha <- NA

for (a in names(cv_results)) {
  err <- min(cv_results[[a]]$cvm)
  if (err < lowest_error) {
    lowest_error <- err
    best_model <- cv_results[[a]]
    best_alpha <- as.numeric(gsub("alpha_", "", a))
  }
}

cat("Best alpha:", best_alpha, "\n")
```

```
## Best alpha: 0.6
```

```
best_lambda <- best_model$lambda.min
```

```
final_model <- glmnet(
  x_train, y_train,
  family = "multinomial",
  lambda = best_lambda
)
```

Evaluating with Cross-Validation:

```
cv_fit <- train(
  x = x_train, y = y_train,
  method = "glmnet",
  family = "multinomial",
  trControl = trainControl(method = "cv", number = 10),
  tuneGrid = expand.grid(alpha = best_alpha, lambda = best_lambda)
)
print(cv_fit)
```

```
## glmnet
##
## 8427 samples
##   12 predictor
##    3 classes: 'Budget Shopper', 'Premium Shopper', 'Regular Shopper'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 7584, 7583, 7584, 7583, 7584, 7585, ...
```

```
## Resampling results:
##
##   Accuracy   Kappa
##   0.3327411  -0.001967025
##
## Tuning parameter 'alpha' was held constant at a value of 0.6
## Tuning
##  parameter 'lambda' was held constant at a value of 0.006942871
```

Results and Discussion

This model was developed to classify customers into different segments based on their demographic and shopping behavior. The dataset includes details like Age, Annual Income, Gender, Product Category Purchased, Average Spend per Visit, Number of Visits in the Last 6 Months, and the target variable Customer Segment (with three categories: Budget Shopper, Regular Shopper, Premium Shopper)

Based from the results, more fashion purchases slightly increased the chance of being a Premium or Regular Shopper.

Buying Books was negatively linked to being a Premium or Regular Shopper.

Gender, Age, and Annual Income had only minor effects on predicting customer segments.