Final\_Case Assignment

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(lubridate)  
library(janitor)

##   
## Attaching package: 'janitor'  
##   
## The following objects are masked from 'package:stats':  
##   
## chisq.test, fisher.test

library(ggplot2)  
library(corrplot)

## corrplot 0.95 loaded

library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(gridExtra)

##   
## Attaching package: 'gridExtra'  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine

library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(rpart)  
library(rpart.plot)  
Veggiemingle <- read\_csv("/cloud/project/6760765/Datasets/Veggiemingle.csv")%>%  
 janitor::clean\_names()

## Rows: 5000 Columns: 33  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (24): Gender, Location, Diet Type, Reason for Diet, Dietary Transition, ...  
## dbl (9): User ID, Age, Diet Duration (years), Cooking Skill, Relationship H...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

head(Veggiemingle)

## # A tibble: 6 × 33  
## user\_id age gender location diet\_type diet\_duration\_years reason\_for\_diet  
## <dbl> <dbl> <chr> <chr> <chr> <dbl> <chr>   
## 1 1 48 Female Dallas Pescatarian 10 Ethical   
## 2 2 32 Female Chicago Pescatarian 12 Health   
## 3 3 31 Male New York Vegan 8 Ethical   
## 4 4 20 Female San Jose Vegan 4 Environmental   
## 5 5 31 Female Dallas Vegetarian 3 Health   
## 6 6 42 Female Houston Vegan 11 Health   
## # ℹ 26 more variables: dietary\_transition <chr>, favorite\_cuisine <chr>,  
## # cooking\_skill <dbl>, exercise\_habits <chr>, favorite\_activity <chr>,  
## # alcohol\_consumption <chr>, smoking\_habits <chr>, education\_level <chr>,  
## # occupation <chr>, income\_level <chr>, relationship\_history <dbl>,  
## # children <chr>, pets <chr>, travel\_preference <chr>, music\_taste <chr>,  
## # movie\_preference <chr>, reading\_habits <dbl>, online\_activity <dbl>,  
## # app\_usage\_frequency <chr>, event\_attendance <chr>, referral\_source <chr>, …

# Preprocess the Veggiemingle dataset  
Veggiemingle\_df <- Veggiemingle %>%  
 # Convert categorical variables to factors  
 mutate(  
 gender = as.factor(gender),  
 location = as.factor(location),  
 diet\_type = as.factor(diet\_type),  
 reason\_for\_diet = as.factor(reason\_for\_diet),  
 dietary\_transition = as.factor(dietary\_transition),  
 favorite\_cuisine = as.factor(favorite\_cuisine),  
 exercise\_habits = as.factor(exercise\_habits),  
 favorite\_activity = as.factor(favorite\_activity),  
 alcohol\_consumption = as.factor(alcohol\_consumption),  
 smoking\_habits = as.factor(smoking\_habits),  
 education\_level = as.factor(education\_level),  
 occupation = as.factor(occupation),  
 income\_level = as.factor(income\_level),  
 children = as.factor(children),  
 pets = as.factor(pets),  
 travel\_preference = as.factor(travel\_preference),  
 music\_taste = as.factor(music\_taste),  
 movie\_preference = as.factor(movie\_preference),  
 app\_usage\_frequency = as.factor(app\_usage\_frequency),  
 event\_attendance = as.factor(event\_attendance),  
 referral\_source = as.factor(referral\_source),  
 response\_rate = as.factor(response\_rate),  
 safety\_concerns = as.factor(safety\_concerns),  
 matched = as.factor(matched)  
 ) %>%  
 # Convert numeric variables to numeric types (if necessary)  
 mutate(  
 age = as.numeric(age),  
 diet\_duration\_years = as.numeric(diet\_duration\_years),  
 cooking\_skill = as.numeric(cooking\_skill),  
 relationship\_history = as.numeric(relationship\_history),  
 reading\_habits = as.numeric(reading\_habits),  
 online\_activity = as.numeric(online\_activity),  
 account\_duration = as.numeric(account\_duration),  
 profile\_completeness = as.numeric(profile\_completeness)  
 )  
  
# Inspect the preprocessed dataset  
glimpse(Veggiemingle\_df)

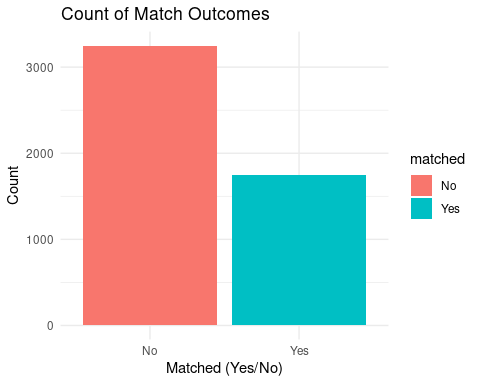
## Rows: 5,000  
## Columns: 33  
## $ user\_id <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15…  
## $ age <dbl> 48, 32, 31, 20, 31, 42, 43, 44, 22, 44, 45, 26, 4…  
## $ gender <fct> Female, Female, Male, Female, Female, Female, Fem…  
## $ location <fct> Dallas, Chicago, New York, San Jose, Dallas, Hous…  
## $ diet\_type <fct> Pescatarian, Pescatarian, Vegan, Vegan, Vegetaria…  
## $ diet\_duration\_years <dbl> 10, 12, 8, 4, 3, 11, 13, 19, 19, 5, 3, 5, 9, 10, …  
## $ reason\_for\_diet <fct> Ethical, Health, Ethical, Environmental, Health, …  
## $ dietary\_transition <fct> Gradual, Gradual, In Phases, In Phases, In Phases…  
## $ favorite\_cuisine <fct> Western, Mediterranean, Indian, Asian, Asian, Med…  
## $ cooking\_skill <dbl> 9, 9, 6, 9, 5, 4, 5, 8, 7, 9, 5, 10, 5, 4, 8, 2, …  
## $ exercise\_habits <fct> Rarely, Rarely, Rarely, Daily, Daily, Rarely, Wee…  
## $ favorite\_activity <fct> Reading, Yoga, Reading, Music, Reading, Hiking, R…  
## $ alcohol\_consumption <fct> Socially, Regularly, Socially, Socially, Socially…  
## $ smoking\_habits <fct> Regularly, Regularly, Socially, Regularly, Social…  
## $ education\_level <fct> Bachelor's, High School, Bachelor's, Bachelor's, …  
## $ occupation <fct> Entrepreneur, Teacher, Engineer, Artist, Entrepre…  
## $ income\_level <fct> High, High, Middle, Middle, Low, High, High, Midd…  
## $ relationship\_history <dbl> 4, 5, 5, 5, 1, 5, 0, 5, 0, 4, 5, 3, 5, 2, 5, 1, 0…  
## $ children <fct> No, No, Yes, No, No, Yes, No, No, No, No, No, No,…  
## $ pets <fct> No, Yes, No, No, No, No, Yes, Yes, No, No, No, Ye…  
## $ travel\_preference <fct> Adventurous, Rarely Travel, Relaxing, Rarely Trav…  
## $ music\_taste <fct> Pop, Pop, Classical, Pop, Jazz, Rock, Pop, Jazz, …  
## $ movie\_preference <fct> Action, Action, Action, Romance, Documentary, Rom…  
## $ reading\_habits <dbl> 14, 9, 14, 11, 18, 10, 10, 4, 4, 7, 13, 18, 15, 1…  
## $ online\_activity <dbl> 2, 1, 10, 9, 4, 5, 6, 2, 3, 9, 9, 8, 4, 4, 2, 1, …  
## $ app\_usage\_frequency <fct> Monthly, Monthly, Weekly, Weekly, Weekly, Daily, …  
## $ event\_attendance <fct> Never, Never, Rarely, Never, Often, Often, Often,…  
## $ referral\_source <fct> Online Ad, Social Media, Friends, Online Ad, Soci…  
## $ account\_duration <dbl> 5, 1, 4, 11, 3, 22, 12, 6, 17, 15, 24, 23, 11, 11…  
## $ profile\_completeness <dbl> 54, 68, 54, 67, 86, 65, 97, 54, 56, 76, 55, 66, 8…  
## $ response\_rate <fct> Always, Rarely, Always, Often, Always, Rarely, Of…  
## $ safety\_concerns <fct> No, No, Yes, No, No, No, No, No, No, No, Yes, No,…  
## $ matched <fct> No, Yes, Yes, No, Yes, No, Yes, No, No, No, No, N…

summary(Veggiemingle\_df)

## user\_id age gender location   
## Min. : 1 Min. :18.00 Female:2424 Dallas : 526   
## 1st Qu.:1251 1st Qu.:26.00 Male :2395 Chicago : 521   
## Median :2500 Median :34.00 Other : 181 Los Angeles : 516   
## Mean :2500 Mean :34.04 Philadelphia: 507   
## 3rd Qu.:3750 3rd Qu.:42.00 Houston : 506   
## Max. :5000 Max. :50.00 San Diego : 505   
## (Other) :1919   
## diet\_type diet\_duration\_years reason\_for\_diet dietary\_transition  
## Pescatarian:1249 Min. : 1.00 Environmental:1235 Gradual :1694   
## Raw Vegan :1295 1st Qu.: 5.00 Ethical :1253 In Phases:1655   
## Vegan :1211 Median :10.00 Health :1271 Overnight:1651   
## Vegetarian :1245 Mean :10.41 Religious :1241   
## 3rd Qu.:15.00   
## Max. :20.00   
##   
## favorite\_cuisine cooking\_skill exercise\_habits favorite\_activity  
## Asian :1312 Min. : 1.000 Daily :1655 Hiking :1280   
## Indian :1184 1st Qu.: 3.000 Rarely:1682 Music :1234   
## Mediterranean:1291 Median : 6.000 Weekly:1663 Reading:1208   
## Western :1213 Mean : 5.567 Yoga :1278   
## 3rd Qu.: 8.000   
## Max. :10.000   
##   
## alcohol\_consumption smoking\_habits education\_level occupation   
## Never :1654 Never :1687 Bachelor's :1234 Artist : 953   
## Regularly:1724 Regularly:1675 High School:1256 Doctor :1040   
## Socially :1622 Socially :1638 Master's :1278 Engineer :1054   
## PhD :1232 Entrepreneur: 969   
## Teacher : 984   
##   
##   
## income\_level relationship\_history children pets travel\_preference  
## High :1664 Min. :0.000 No :3450 No :2529 Adventurous :1213   
## Low :1766 1st Qu.:1.000 Yes:1550 Yes:2471 Cultural :1253   
## Middle:1570 Median :3.000 Rarely Travel:1262   
## Mean :2.504 Relaxing :1272   
## 3rd Qu.:4.000   
## Max. :5.000   
##   
## music\_taste movie\_preference reading\_habits online\_activity   
## Classical:1215 Action :1258 Min. : 1.00 Min. : 1.000   
## Jazz :1218 Comedy :1219 1st Qu.: 6.00 1st Qu.: 3.000   
## Pop :1268 Documentary:1283 Median :10.00 Median : 5.000   
## Rock :1299 Romance :1240 Mean :10.51 Mean : 5.452   
## 3rd Qu.:16.00 3rd Qu.: 8.000   
## Max. :20.00 Max. :10.000   
##   
## app\_usage\_frequency event\_attendance referral\_source account\_duration  
## Daily :1691 Never :1685 Friends :1648 Min. : 1.00   
## Monthly:1634 Often :1658 Online Ad :1639 1st Qu.: 7.00   
## Weekly :1675 Rarely:1657 Social Media:1713 Median :13.00   
## Mean :12.52   
## 3rd Qu.:18.00   
## Max. :24.00   
##   
## profile\_completeness response\_rate safety\_concerns matched   
## Min. : 50.00 Always:1650 No :4516 No :3250   
## 1st Qu.: 62.00 Often :1663 Yes: 484 Yes:1750   
## Median : 75.00 Rarely:1687   
## Mean : 74.98   
## 3rd Qu.: 87.25   
## Max. :100.00   
##

#Expolatory Data Analysis

# 1. Distribution of the target variable (Matched)  
library(ggplot2)  
ggplot(Veggiemingle\_df, aes(x = matched, fill = matched)) +  
 geom\_bar() +  
 labs(title = "Count of Match Outcomes", x = "Matched (Yes/No)", y = "Count") +  
 theme\_minimal()

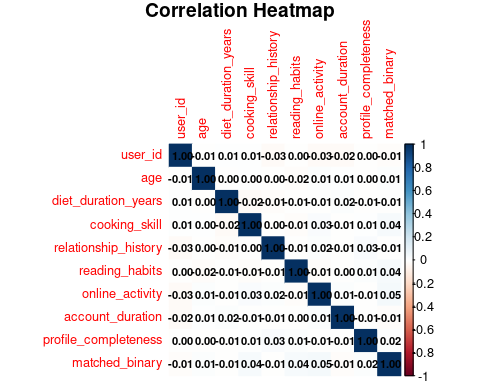


# Scale numeric features before PCA  
numeric\_vars <- Veggiemingle\_df %>% select(where(is.numeric))  
numeric\_vars\_scaled <- scale(numeric\_vars)

## Correlation Heatmap for Numeric Variables  
# Load required libraries  
library(dplyr)  
library(corrplot)  
  
# Ensure dataset is loaded  
# Replace `dataset` with your actual dataset name  
# Assume `matched\_binary` is the target variable (0/1 binary)  
  
# Select numeric variables (including target variable)  
numeric\_vars <- Veggiemingle\_df %>%  
 select(where(is.numeric)) # Select all numeric columns  
  
# Check if target variable is included  
if (!"matched\_binary" %in% colnames(numeric\_vars)) {  
 numeric\_vars <- Veggiemingle\_df %>%  
 mutate(matched\_binary = as.numeric(as.factor(matched))) %>% # Convert target variable if needed  
 select(where(is.numeric))  
}  
  
# Calculate the correlation matrix  
correlation\_matrix <- cor(numeric\_vars, use = "complete.obs")  
  
# Display the correlation matrix  
print(correlation\_matrix)

## user\_id age diet\_duration\_years  
## user\_id 1.000000000 -0.007148380 0.008839968  
## age -0.007148380 1.000000000 -0.001289270  
## diet\_duration\_years 0.008839968 -0.001289270 1.000000000  
## cooking\_skill 0.009188582 0.002866800 -0.021657809  
## relationship\_history -0.029557538 -0.003033498 -0.012957482  
## reading\_habits -0.003683153 -0.016002992 -0.005642905  
## online\_activity -0.027789569 0.011735366 -0.005339430  
## account\_duration -0.020323013 0.005337011 0.016094205  
## profile\_completeness -0.002001775 -0.003261357 -0.009290716  
## matched\_binary -0.010670694 0.011428414 -0.006220968  
## cooking\_skill relationship\_history reading\_habits  
## user\_id 0.009188582 -0.029557538 -0.0036831531  
## age 0.002866800 -0.003033498 -0.0160029924  
## diet\_duration\_years -0.021657809 -0.012957482 -0.0056429046  
## cooking\_skill 1.000000000 -0.001176148 -0.0130817297  
## relationship\_history -0.001176148 1.000000000 -0.0067145366  
## reading\_habits -0.013081730 -0.006714537 1.0000000000  
## online\_activity 0.026901610 0.015853098 -0.0139559461  
## account\_duration -0.008109303 -0.005775613 -0.0006813267  
## profile\_completeness 0.005589198 0.028761319 0.0130610128  
## matched\_binary 0.035747187 -0.008026570 0.0388503808  
## online\_activity account\_duration profile\_completeness  
## user\_id -0.027789569 -0.0203230127 -0.002001775  
## age 0.011735366 0.0053370115 -0.003261357  
## diet\_duration\_years -0.005339430 0.0160942051 -0.009290716  
## cooking\_skill 0.026901610 -0.0081093031 0.005589198  
## relationship\_history 0.015853098 -0.0057756134 0.028761319  
## reading\_habits -0.013955946 -0.0006813267 0.013061013  
## online\_activity 1.000000000 0.0147991400 -0.007260913  
## account\_duration 0.014799140 1.0000000000 -0.009366352  
## profile\_completeness -0.007260913 -0.0093663522 1.000000000  
## matched\_binary 0.046560541 -0.0112524586 0.018811556  
## matched\_binary  
## user\_id -0.010670694  
## age 0.011428414  
## diet\_duration\_years -0.006220968  
## cooking\_skill 0.035747187  
## relationship\_history -0.008026570  
## reading\_habits 0.038850381  
## online\_activity 0.046560541  
## account\_duration -0.011252459  
## profile\_completeness 0.018811556  
## matched\_binary 1.000000000

# Plot the correlation heatmap  
corrplot(  
 correlation\_matrix,  
 method = "color",  
 tl.cex = 0.8,  
 number.cex = 0.7,  
 addCoef.col = "black",  
 title = "Correlation Heatmap",  
 mar = c(0, 0, 1, 0)  
)



# Extract correlations with the target variable  
target\_correlations <- correlation\_matrix["matched\_binary", ]  
  
# Display correlations with the target variable  
print(target\_correlations)

## user\_id age diet\_duration\_years   
## -0.010670694 0.011428414 -0.006220968   
## cooking\_skill relationship\_history reading\_habits   
## 0.035747187 -0.008026570 0.038850381   
## online\_activity account\_duration profile\_completeness   
## 0.046560541 -0.011252459 0.018811556   
## matched\_binary   
## 1.000000000

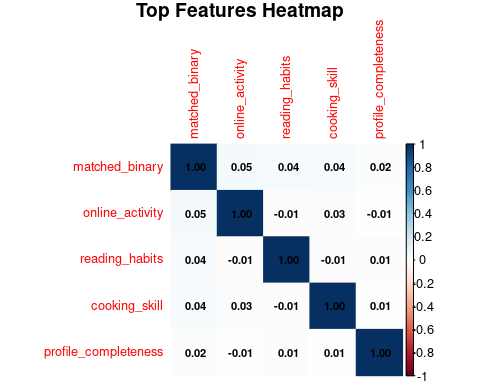
# Extract and display top correlated features with the target variable  
top\_correlations <- sort(abs(target\_correlations), decreasing = TRUE)  
print("Top correlated features with 'matched\_binary':")

## [1] "Top correlated features with 'matched\_binary':"

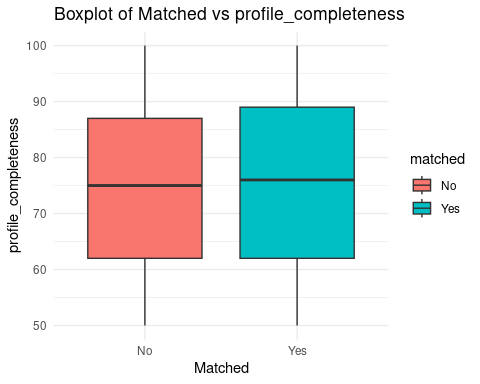
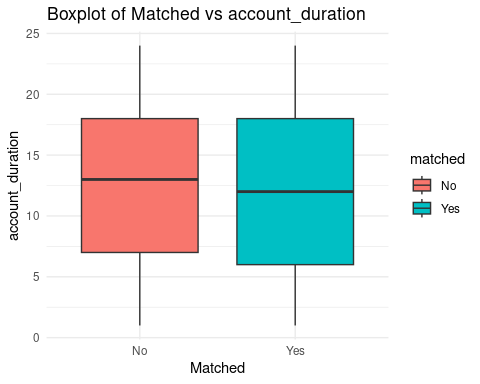
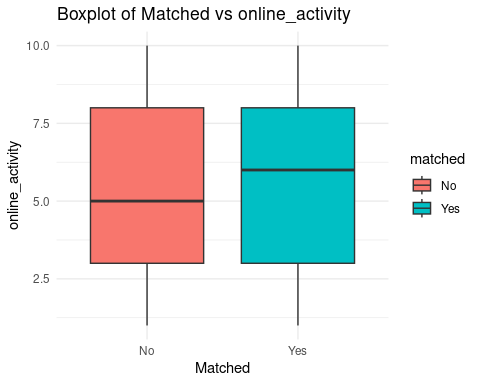
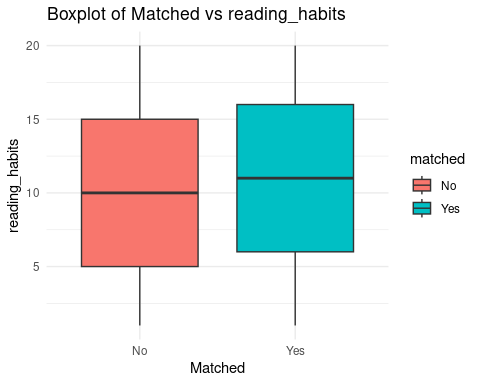
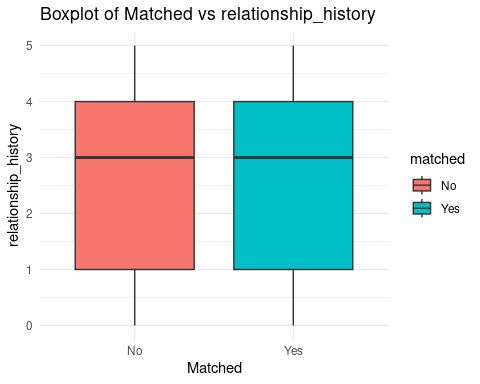
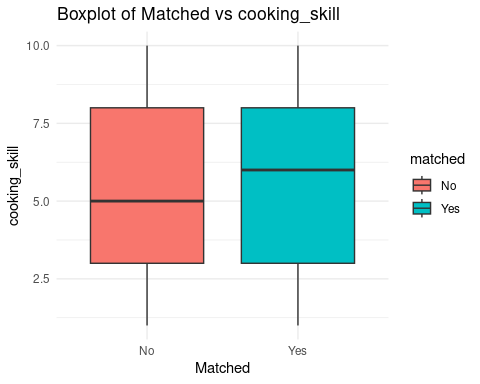
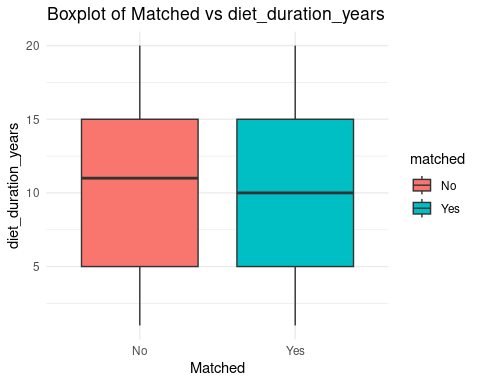
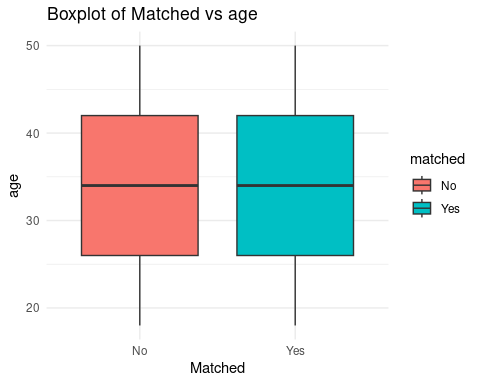
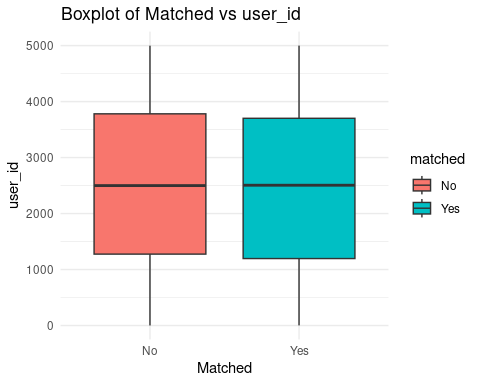
print(top\_correlations)

## matched\_binary online\_activity reading\_habits   
## 1.000000000 0.046560541 0.038850381   
## cooking\_skill profile\_completeness age   
## 0.035747187 0.018811556 0.011428414   
## account\_duration user\_id relationship\_history   
## 0.011252459 0.010670694 0.008026570   
## diet\_duration\_years   
## 0.006220968

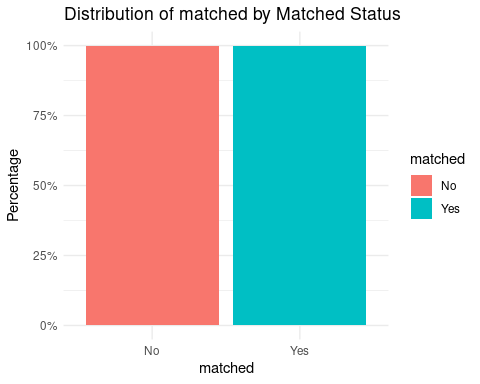
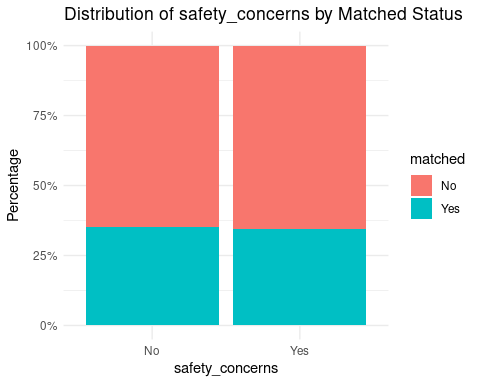
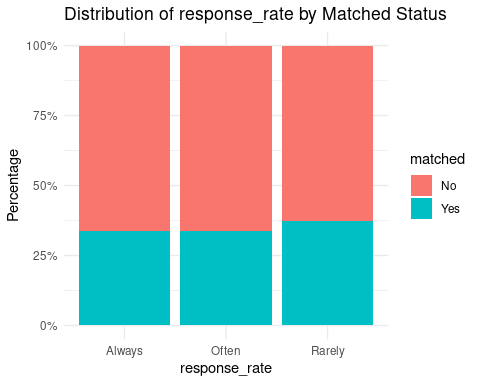
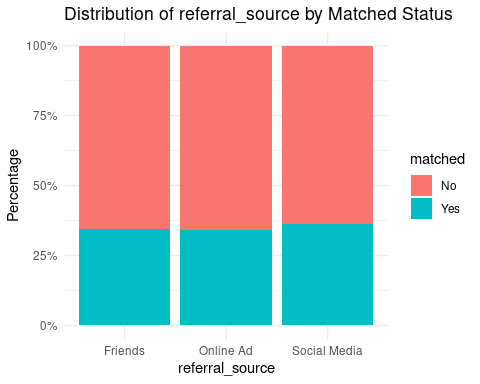
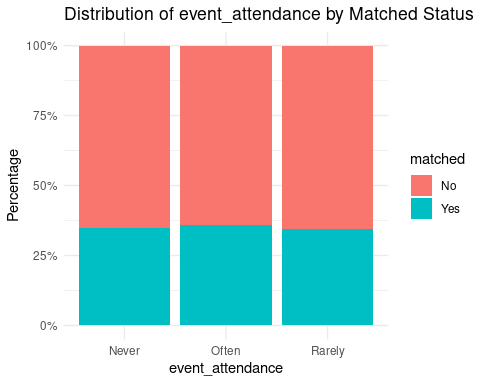
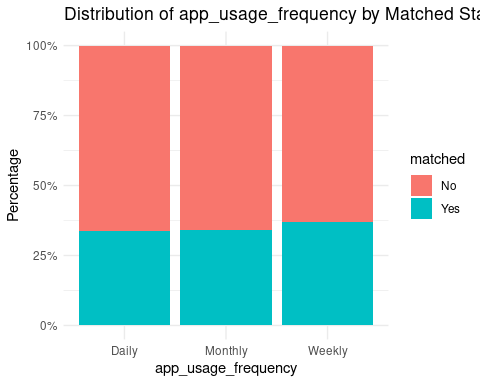
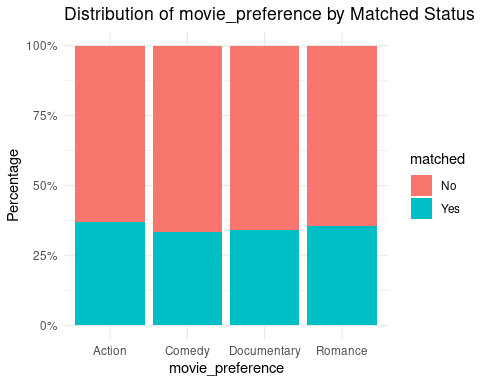
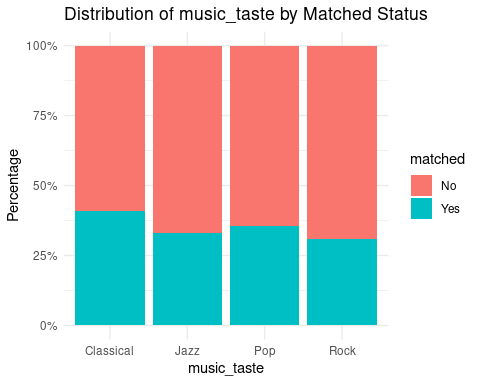
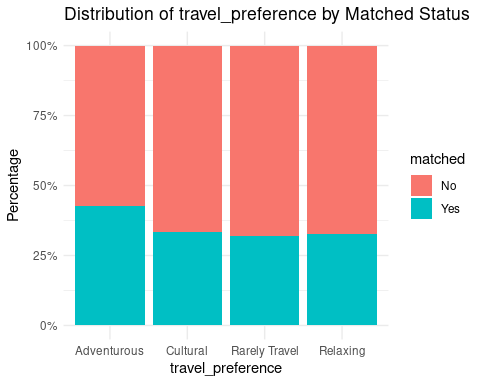
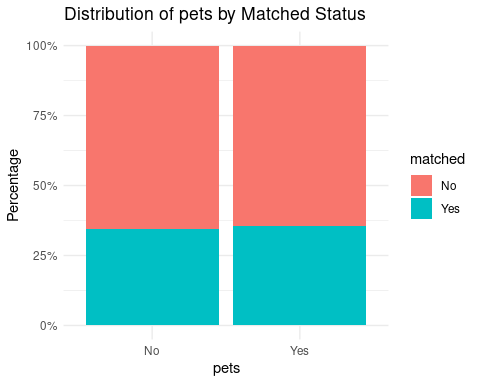
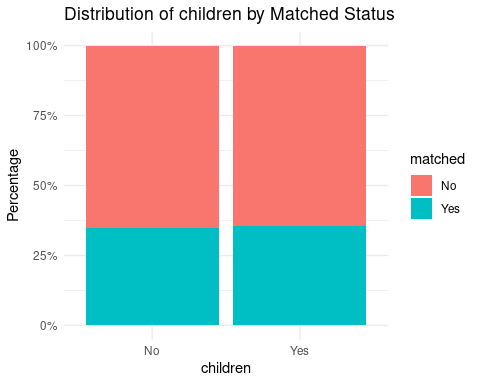
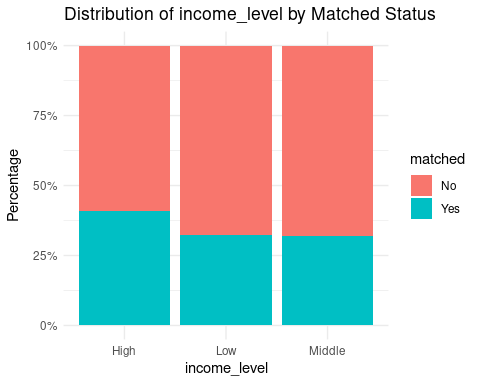
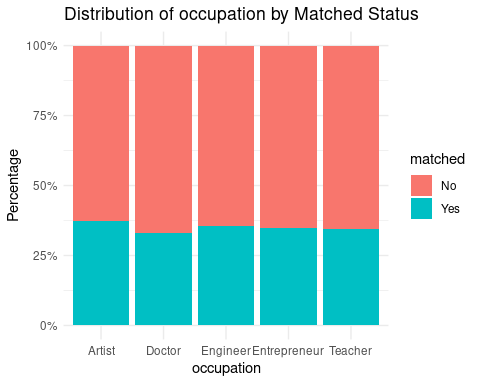
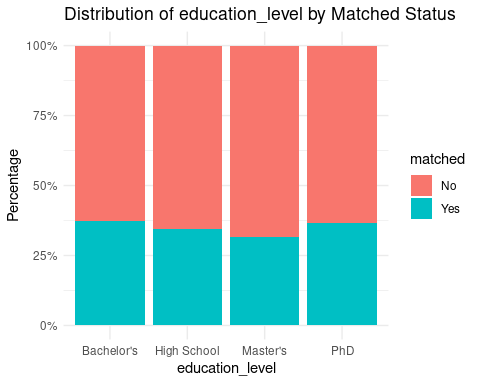
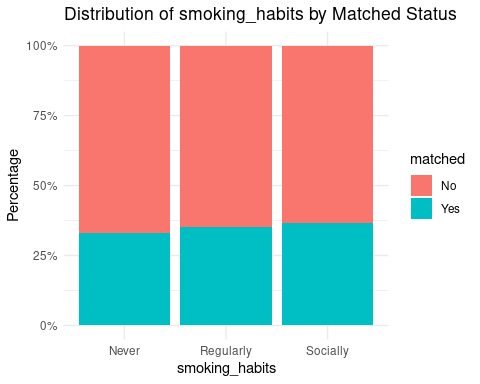
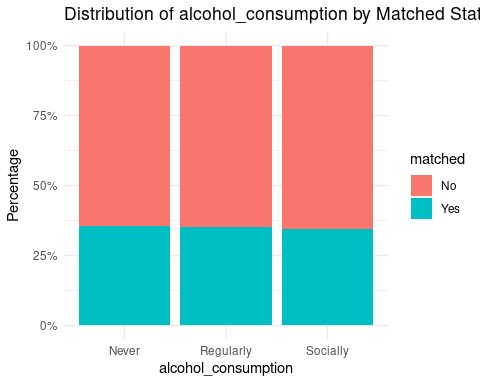
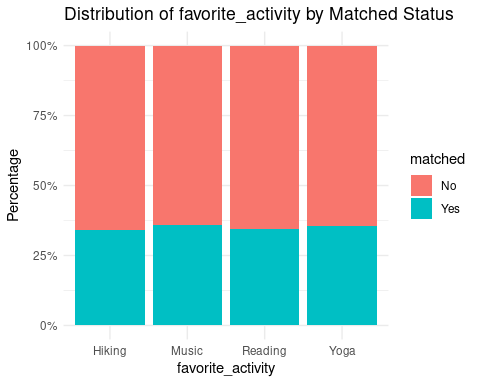
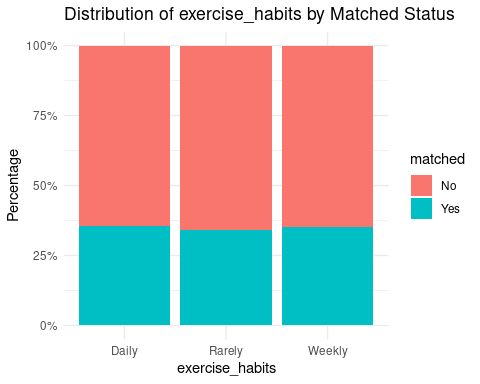
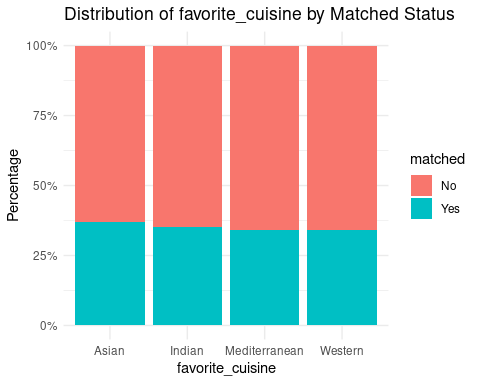
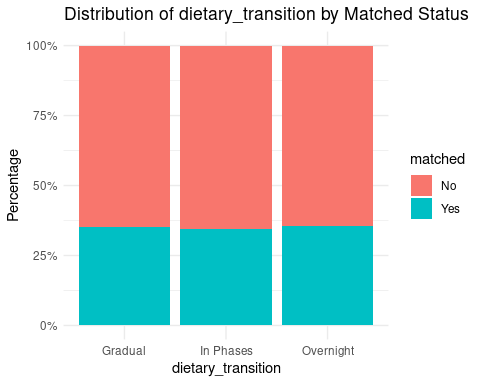
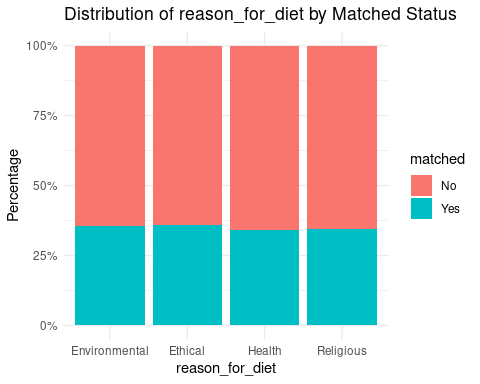
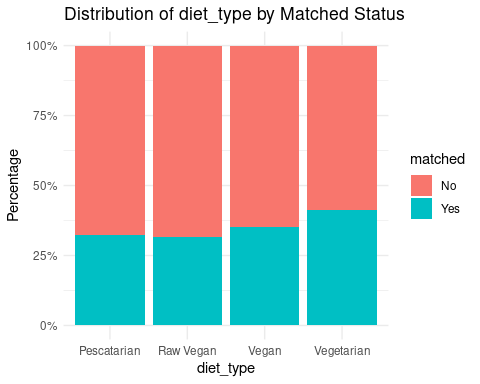
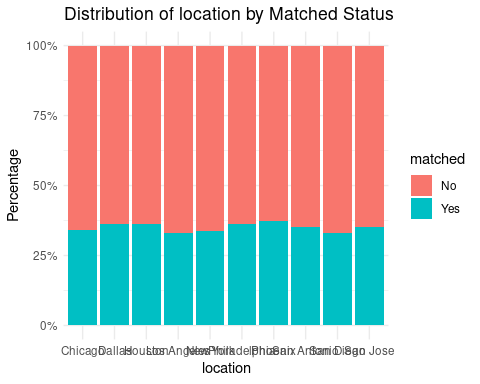
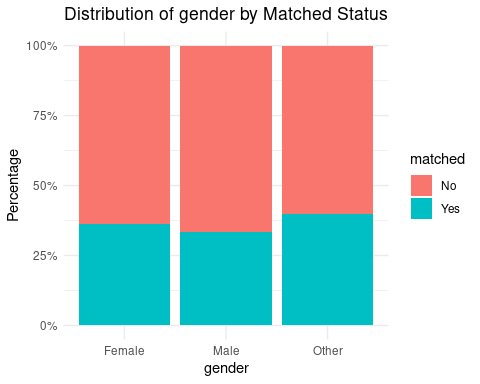
# Optionally plot the top correlated features (if required)  
# Filter for top 5 correlations for simplicity  
top\_features <- names(sort(target\_correlations, decreasing = TRUE)[1:5])  
heatmap\_matrix <- correlation\_matrix[top\_features, top\_features]  
  
corrplot(  
 heatmap\_matrix,  
 method = "color",  
 tl.cex = 0.8,  
 number.cex = 0.7,  
 addCoef.col = "black",  
 title = "Top Features Heatmap",  
 mar = c(0, 0, 1, 0)  
)



# Boxplot of Matched vs Numeric Variables  
library(ggplot2)  
  
# Define categorical and numeric variables  
categorical\_vars <- names(Veggiemingle\_df)[sapply(Veggiemingle\_df, is.factor)]  
numeric\_vars <- names(Veggiemingle\_df)[sapply(Veggiemingle\_df, is.numeric)]  
  
# Check if the 'matched' column exists and is a factor  
if (!"matched" %in% colnames(Veggiemingle\_df)) {  
 stop("The 'matched' column is missing from the dataset.")  
}  
  
if (!is.factor(Veggiemingle\_df$matched)) {  
 Veggiemingle\_df$matched <- as.factor(Veggiemingle\_df$matched)  
}  
  
# Boxplots for numeric variables  
for (col in numeric\_vars) {  
 p <- ggplot(Veggiemingle\_df, aes(x = matched, y = .data[[col]], fill = matched)) +  
 geom\_boxplot() +  
 theme\_minimal() +  
 labs(  
 title = paste("Boxplot of Matched vs", col),  
 x = "Matched",  
 y = col  
 )  
 print(p)  
}



# Bar plots for categorical variables  
for (col in categorical\_vars) {  
 p <- ggplot(Veggiemingle\_df, aes(x = .data[[col]], fill = matched)) +  
 geom\_bar(position = "fill") +  
 theme\_minimal() +  
 scale\_y\_continuous(labels = scales::percent) +  
 labs(  
 title = paste("Distribution of", col, "by Matched Status"),  
 x = col,  
 y = "Percentage"  
 )  
 print(p)  
}

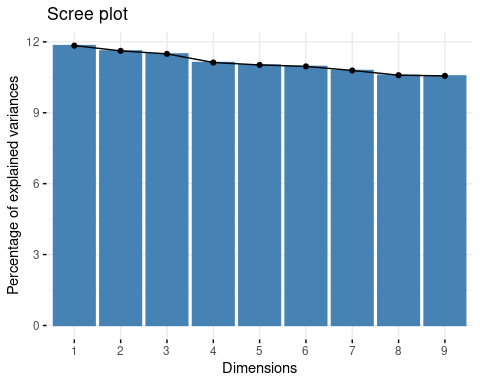


#Principal Componet analysis

# Perform PCA  
  
pca\_result <- prcomp(numeric\_vars\_scaled, center = TRUE, scale. = TRUE)  
summary(pca\_result)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8  
## Standard deviation 1.0323 1.0226 1.0170 1.0007 0.9960 0.9932 0.9854 0.9762  
## Proportion of Variance 0.1184 0.1162 0.1149 0.1113 0.1102 0.1096 0.1079 0.1059  
## Cumulative Proportion 0.1184 0.2346 0.3495 0.4608 0.5710 0.6806 0.7885 0.8944  
## PC9  
## Standard deviation 0.9749  
## Proportion of Variance 0.1056  
## Cumulative Proportion 1.0000

fviz\_eig(pca\_result)



# Determine number of components explaining at least 80% variance  
explained\_variance <- summary(pca\_result)$importance[2, ]  
cumulative\_variance <- cumsum(explained\_variance)  
num\_components <- which(cumulative\_variance >= 0.8)[1]  
pca\_components <- pca\_result$x[, 1:num\_components]  
Veggiemingle\_df <- cbind(Veggiemingle\_df, pca\_components)

#Model Building

# Ensure the target variable 'matched' is available  
if (!"matched" %in% colnames(Veggiemingle\_df)) {  
 stop("The 'matched' column is missing from the dataset.")  
}  
  
# Check if 'matched' is a factor  
if (!is.factor(Veggiemingle\_df$matched)) {  
 Veggiemingle\_df$matched <- as.factor(Veggiemingle\_df$matched)  
}  
  
# Select only the first two principal components (PC1 and PC2)  
pca\_data <- cbind(data.frame(pca\_result$x[, 1:2]), matched = Veggiemingle\_df$matched)  
  
# Split the PCA data into training and testing sets  
set.seed(123) # For reproducibility  
train\_indices <- createDataPartition(pca\_data$matched, p = 0.8, list = FALSE)  
train\_pca <- pca\_data[train\_indices, ]  
test\_pca <- pca\_data[-train\_indices, ]  
  
# -------------------------------------------------  
# Logistic Regression with PCA Components (PC1 and PC2)  
# -------------------------------------------------  
logistic\_model <- glm(matched ~ ., data = train\_pca, family = "binomial")  
summary(logistic\_model)

##   
## Call:  
## glm(formula = matched ~ ., family = "binomial", data = train\_pca)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.61988 0.03318 -18.681 <2e-16 \*\*\*  
## PC1 -0.07006 0.03216 -2.178 0.0294 \*   
## PC2 -0.03123 0.03249 -0.961 0.3364   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 5179.6 on 3999 degrees of freedom  
## Residual deviance: 5173.9 on 3997 degrees of freedom  
## AIC: 5179.9  
##   
## Number of Fisher Scoring iterations: 4

# Predict on the test set  
logistic\_preds <- predict(logistic\_model, test\_pca, type = "response")  
logistic\_pred\_classes <- ifelse(logistic\_preds > 0.5, "Yes", "No")  
  
# Confusion Matrix for Logistic Regression  
confusion\_matrix\_logistic <- confusionMatrix(as.factor(logistic\_pred\_classes), test\_pca$matched)

## Warning in confusionMatrix.default(as.factor(logistic\_pred\_classes),  
## test\_pca$matched): Levels are not in the same order for reference and data.  
## Refactoring data to match.

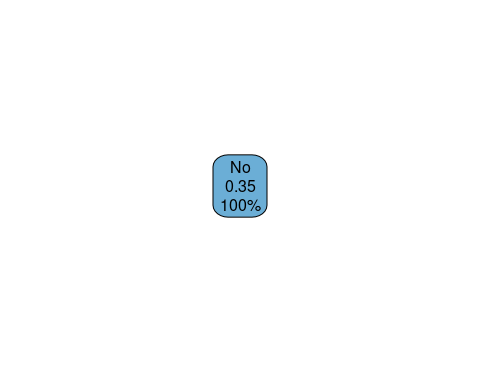
print(confusion\_matrix\_logistic)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 650 350  
## Yes 0 0  
##   
## Accuracy : 0.65   
## 95% CI : (0.6195, 0.6796)  
## No Information Rate : 0.65   
## P-Value [Acc > NIR] : 0.5145   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 1.00   
## Specificity : 0.00   
## Pos Pred Value : 0.65   
## Neg Pred Value : NaN   
## Prevalence : 0.65   
## Detection Rate : 0.65   
## Detection Prevalence : 1.00   
## Balanced Accuracy : 0.50   
##   
## 'Positive' Class : No   
##

# -------------------------------------------------  
# KNN Classifier with PCA Components (PC1 and PC2)  
# -------------------------------------------------  
knn\_model <- train(matched ~ ., data = train\_pca, method = "knn", tuneLength = 5)  
knn\_pred <- predict(knn\_model, test\_pca)  
confusion\_matrix\_knn <- confusionMatrix(knn\_pred, test\_pca$matched)  
print(confusion\_matrix\_knn)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 557 305  
## Yes 93 45  
##   
## Accuracy : 0.602   
## 95% CI : (0.5709, 0.6325)  
## No Information Rate : 0.65   
## P-Value [Acc > NIR] : 0.9993   
##   
## Kappa : -0.0169   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.8569   
## Specificity : 0.1286   
## Pos Pred Value : 0.6462   
## Neg Pred Value : 0.3261   
## Prevalence : 0.6500   
## Detection Rate : 0.5570   
## Detection Prevalence : 0.8620   
## Balanced Accuracy : 0.4927   
##   
## 'Positive' Class : No   
##

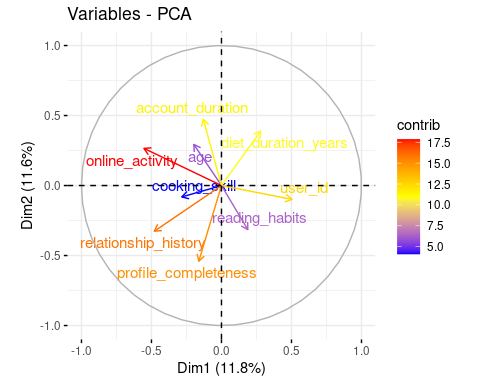
# -------------------------------------------------  
# Decision Tree with PCA Components (PC1 and PC2)  
# -------------------------------------------------  
decision\_tree\_model <- rpart(matched ~ ., data = train\_pca, method = "class")  
rpart.plot(decision\_tree\_model)



# Predict on the test set  
tree\_preds <- predict(decision\_tree\_model, test\_pca, type = "class")  
confusion\_matrix\_tree <- confusionMatrix(tree\_preds, test\_pca$matched)  
print(confusion\_matrix\_tree)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 650 350  
## Yes 0 0  
##   
## Accuracy : 0.65   
## 95% CI : (0.6195, 0.6796)  
## No Information Rate : 0.65   
## P-Value [Acc > NIR] : 0.5145   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 1.00   
## Specificity : 0.00   
## Pos Pred Value : 0.65   
## Neg Pred Value : NaN   
## Prevalence : 0.65   
## Detection Rate : 0.65   
## Detection Prevalence : 1.00   
## Balanced Accuracy : 0.50   
##   
## 'Positive' Class : No   
##

# -------------------------------------------------  
# Insights and Visualization  
# -------------------------------------------------  
  
# PCA Component Contribution (PC1 and PC2 only)  
fviz\_pca\_var(pca\_result, axes = c(1, 2), col.var = "contrib", gradient.cols = c("blue", "yellow", "red"), repel = TRUE)



# Extract the confusion matrix table  
conf\_matrix <- confusion\_matrix\_knn$table  
  
# Check the structure of the confusion matrix  
str(conf\_matrix)

## 'table' int [1:2, 1:2] 557 93 305 45  
## - attr(\*, "dimnames")=List of 2  
## ..$ Prediction: chr [1:2] "No" "Yes"  
## ..$ Reference : chr [1:2] "No" "Yes"

# Convert to a data frame for ggplot2  
conf\_matrix\_df <- as.data.frame(conf\_matrix)  
  
# Check column names to ensure the correct names for ggplot  
colnames(conf\_matrix\_df)

## [1] "Prediction" "Reference" "Freq"

# Plot the confusion matrix heatmap  
ggplot(conf\_matrix\_df, aes(x = Reference, y = Prediction, fill = Freq)) +  
 geom\_tile() +  
 scale\_fill\_gradient(low = "white", high = "steelblue") +  
 geom\_text(aes(label = Freq), vjust = 1) +  
 theme\_minimal() +  
 labs(title = "Confusion Matrix Heatmap", x = "Predicted", y = "Actual") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))

