Mental Health Analytics in Work Place

Jayneet Jain,Mihika kubadia,Spoorthy Nagendra

2024-10-03

# Load necessary libraries  
if (!require("pacman"))   
 install.packages("pacman")

## Loading required package: pacman

pacman::p\_load(ggplot2,  
 factoextra,  
 cluster,  
 dendextend,  
 arules,  
 arulesViz,  
 randomForest,  
 forcats,  
 ROCR,  
 dplyr,  
 caret,  
 randomForest  
   
   
   
)

# Load data  
data <- read.csv("survey.csv")  
# Get the unique data types  
unique\_data\_types <- unique(sapply(data, class))  
  
# Print the unique data types  
print(unique\_data\_types)

## [1] "character" "numeric"

# Examine data structure  
str(data)

## 'data.frame': 1259 obs. of 27 variables:  
## $ Timestamp : chr "2014-08-27 11:29:31" "2014-08-27 11:29:37" "2014-08-27 11:29:44" "2014-08-27 11:29:46" ...  
## $ Age : num 37 44 32 31 31 33 35 39 42 23 ...  
## $ Gender : chr "Female" "M" "Male" "Male" ...  
## $ Country : chr "United States" "United States" "Canada" "United Kingdom" ...  
## $ state : chr "IL" "IN" NA NA ...  
## $ self\_employed : chr NA NA NA NA ...  
## $ family\_history : chr "No" "No" "No" "Yes" ...  
## $ treatment : chr "Yes" "No" "No" "Yes" ...  
## $ work\_interfere : chr "Often" "Rarely" "Rarely" "Often" ...  
## $ no\_employees : chr "6-25" "More than 1000" "6-25" "26-100" ...  
## $ remote\_work : chr "No" "No" "No" "No" ...  
## $ tech\_company : chr "Yes" "No" "Yes" "Yes" ...  
## $ benefits : chr "Yes" "Don't know" "No" "No" ...  
## $ care\_options : chr "Not sure" "No" "No" "Yes" ...  
## $ wellness\_program : chr "No" "Don't know" "No" "No" ...  
## $ seek\_help : chr "Yes" "Don't know" "No" "No" ...  
## $ anonymity : chr "Yes" "Don't know" "Don't know" "No" ...  
## $ leave : chr "Somewhat easy" "Don't know" "Somewhat difficult" "Somewhat difficult" ...  
## $ mental\_health\_consequence: chr "No" "Maybe" "No" "Yes" ...  
## $ phys\_health\_consequence : chr "No" "No" "No" "Yes" ...  
## $ coworkers : chr "Some of them" "No" "Yes" "Some of them" ...  
## $ supervisor : chr "Yes" "No" "Yes" "No" ...  
## $ mental\_health\_interview : chr "No" "No" "Yes" "Maybe" ...  
## $ phys\_health\_interview : chr "Maybe" "No" "Yes" "Maybe" ...  
## $ mental\_vs\_physical : chr "Yes" "Don't know" "No" "No" ...  
## $ obs\_consequence : chr "No" "No" "No" "Yes" ...  
## $ comments : chr NA NA NA NA ...

head(data)

## Timestamp Age Gender Country state self\_employed  
## 1 2014-08-27 11:29:31 37 Female United States IL <NA>  
## 2 2014-08-27 11:29:37 44 M United States IN <NA>  
## 3 2014-08-27 11:29:44 32 Male Canada <NA> <NA>  
## 4 2014-08-27 11:29:46 31 Male United Kingdom <NA> <NA>  
## 5 2014-08-27 11:30:22 31 Male United States TX <NA>  
## 6 2014-08-27 11:31:22 33 Male United States TN <NA>  
## family\_history treatment work\_interfere no\_employees remote\_work  
## 1 No Yes Often 6-25 No  
## 2 No No Rarely More than 1000 No  
## 3 No No Rarely 6-25 No  
## 4 Yes Yes Often 26-100 No  
## 5 No No Never 100-500 Yes  
## 6 Yes No Sometimes 6-25 No  
## tech\_company benefits care\_options wellness\_program seek\_help anonymity  
## 1 Yes Yes Not sure No Yes Yes  
## 2 No Don't know No Don't know Don't know Don't know  
## 3 Yes No No No No Don't know  
## 4 Yes No Yes No No No  
## 5 Yes Yes No Don't know Don't know Don't know  
## 6 Yes Yes Not sure No Don't know Don't know  
## leave mental\_health\_consequence phys\_health\_consequence  
## 1 Somewhat easy No No  
## 2 Don't know Maybe No  
## 3 Somewhat difficult No No  
## 4 Somewhat difficult Yes Yes  
## 5 Don't know No No  
## 6 Don't know No No  
## coworkers supervisor mental\_health\_interview phys\_health\_interview  
## 1 Some of them Yes No Maybe  
## 2 No No No No  
## 3 Yes Yes Yes Yes  
## 4 Some of them No Maybe Maybe  
## 5 Some of them Yes Yes Yes  
## 6 Yes Yes No Maybe  
## mental\_vs\_physical obs\_consequence comments  
## 1 Yes No <NA>  
## 2 Don't know No <NA>  
## 3 No No <NA>  
## 4 No Yes <NA>  
## 5 Don't know No <NA>  
## 6 Don't know No <NA>

summary(data)

## Timestamp Age Gender Country   
## Length:1259 Min. :-1.726e+03 Length:1259 Length:1259   
## Class :character 1st Qu.: 2.700e+01 Class :character Class :character   
## Mode :character Median : 3.100e+01 Mode :character Mode :character   
## Mean : 7.943e+07   
## 3rd Qu.: 3.600e+01   
## Max. : 1.000e+11   
## state self\_employed family\_history treatment   
## Length:1259 Length:1259 Length:1259 Length:1259   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## work\_interfere no\_employees remote\_work tech\_company   
## Length:1259 Length:1259 Length:1259 Length:1259   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## benefits care\_options wellness\_program seek\_help   
## Length:1259 Length:1259 Length:1259 Length:1259   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## anonymity leave mental\_health\_consequence  
## Length:1259 Length:1259 Length:1259   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
## phys\_health\_consequence coworkers supervisor   
## Length:1259 Length:1259 Length:1259   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
## mental\_health\_interview phys\_health\_interview mental\_vs\_physical  
## Length:1259 Length:1259 Length:1259   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
## obs\_consequence comments   
## Length:1259 Length:1259   
## Class :character Class :character   
## Mode :character Mode :character   
##   
##   
##

# Summary statistics for the entire dataset  
summary(data)

## Timestamp Age Gender Country   
## Length:1259 Min. :-1.726e+03 Length:1259 Length:1259   
## Class :character 1st Qu.: 2.700e+01 Class :character Class :character   
## Mode :character Median : 3.100e+01 Mode :character Mode :character   
## Mean : 7.943e+07   
## 3rd Qu.: 3.600e+01   
## Max. : 1.000e+11   
## state self\_employed family\_history treatment   
## Length:1259 Length:1259 Length:1259 Length:1259   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## work\_interfere no\_employees remote\_work tech\_company   
## Length:1259 Length:1259 Length:1259 Length:1259   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## benefits care\_options wellness\_program seek\_help   
## Length:1259 Length:1259 Length:1259 Length:1259   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## anonymity leave mental\_health\_consequence  
## Length:1259 Length:1259 Length:1259   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
## phys\_health\_consequence coworkers supervisor   
## Length:1259 Length:1259 Length:1259   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
## mental\_health\_interview phys\_health\_interview mental\_vs\_physical  
## Length:1259 Length:1259 Length:1259   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
## obs\_consequence comments   
## Length:1259 Length:1259   
## Class :character Class :character   
## Mode :character Mode :character   
##   
##   
##

# Summary statistics for numerical variables  
numeric\_summary <- data %>%  
 select\_if(is.numeric) %>%  
 summarise\_all(list(  
 Mean = ~mean(., na.rm = TRUE),  
 Median = ~median(., na.rm = TRUE),  
 SD = ~sd(., na.rm = TRUE),  
 Min = ~min(., na.rm = TRUE),  
 Max = ~max(., na.rm = TRUE)  
 ))  
print("Summary for numerical variables:")

## [1] "Summary for numerical variables:"

print(numeric\_summary)

## Mean Median SD Min Max  
## 1 79428148 31 2818299443 -1726 1e+11

# Summary statistics for categorical variables  
categorical\_summary <- data %>%  
 select\_if(is.factor) %>%  
 summarise\_all(list(  
 Count = ~n\_distinct(.),  
 Most\_Frequent = ~names(sort(table(.), decreasing = TRUE))[1],  
 Frequency = ~max(table(.))  
 ))  
print("Summary for categorical variables:")

## [1] "Summary for categorical variables:"

print(categorical\_summary)

## data frame with 0 columns and 1 row

# Convert categorical columns to factors  
data$Gender <- as.factor(data$Gender)  
data$family\_history <- as.factor(data$family\_history)  
data$treatment <- as.factor(data$treatment)  
data$work\_interfere <- as.factor(data$work\_interfere)

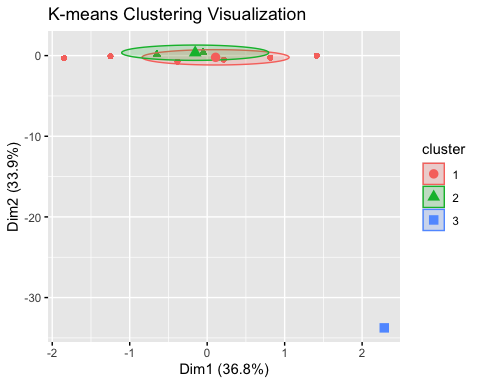
k-means

# Define a mode imputation function  
impute\_mode <- function(x) {  
 mode\_val <- names(sort(table(x), decreasing = TRUE))[1]  
 x[is.na(x)] <- mode\_val  
 return(x)  
}  
  
# Apply mode imputation to work\_interfere  
data$work\_interfere <- impute\_mode(data$work\_interfere)  
summary(data$work\_interfere)

## Never Often Rarely Sometimes   
## 213 144 173 729

# Select and scale relevant features for clustering  
data\_scaled <- data %>%  
 select(Age, work\_interfere, family\_history) %>%  
 mutate\_all(as.numeric) %>% # Convert categorical variables to numeric  
 scale()  
# Perform K-means clustering  
set.seed(123)  
kmeans\_result <- kmeans(data\_scaled, centers = 3) # Change 'centers' as needed  
  
# Add cluster assignments to the original dataset  
data$Cluster <- as.factor(kmeans\_result$cluster)  
  
# Visualize the clusters  
fviz\_cluster(kmeans\_result, data = data\_scaled, geom = "point", ellipse.type = "euclid") +  
 labs(title = "K-means Clustering Visualization")

## Too few points to calculate an ellipse



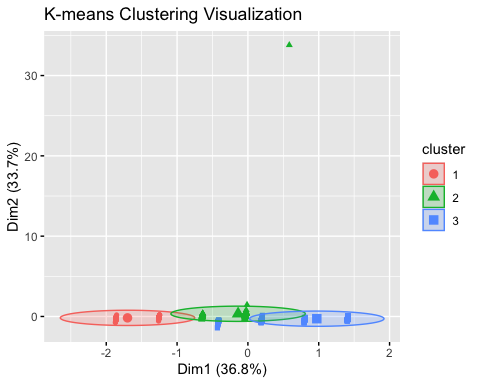
data %>%  
 group\_by(Cluster) %>%  
 summarize(  
 Avg\_Age = mean(Age),  
 Work\_Interfere\_Mode = names(sort(table(work\_interfere), decreasing = TRUE))[1],  
 Family\_History\_Rate = mean(as.numeric(family\_history == "Yes"))  
 )

## # A tibble: 3 × 4  
## Cluster Avg\_Age Work\_Interfere\_Mode Family\_History\_Rate  
## <fct> <dbl> <chr> <dbl>  
## 1 1 32.6 Sometimes 0.673  
## 2 2 28.3 Sometimes 0   
## 3 3 99999999999 Often 1

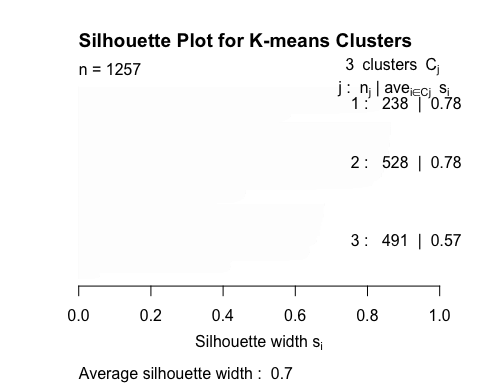
# Visualize Age distribution  
boxplot(data$Age, main = "Age Distribution", horizontal = TRUE)



# Remove rows with extreme outliers  
data <- data %>% filter(Age < 100) # Assuming valid ages are below 100  
  
# Re-scale and re-run clustering  
data\_scaled <- data %>%  
 select(Age, work\_interfere, family\_history) %>%  
 mutate\_all(as.numeric) %>%  
 scale()  
  
kmeans\_result <- kmeans(data\_scaled, centers = 3)  
data$Cluster <- as.factor(kmeans\_result$cluster)  
  
fviz\_cluster(kmeans\_result, data = data\_scaled, geom = "point", ellipse.type = "euclid") +  
 labs(title = "K-means Clustering Visualization")



#now validating the results..  
# Compute silhouette scores  
silhouette\_score <- silhouette(kmeans\_result$cluster, dist(data\_scaled))  
  
# Plot silhouette scores  
plot(silhouette\_score, main = "Silhouette Plot for K-means Clusters")



# Average silhouette width  
cat("Average Silhouette Width:", mean(silhouette\_score[, 3]), "\n")

## Average Silhouette Width: 0.6978305

# Summarize clusters  
cluster\_summary <- data %>%  
 group\_by(Cluster) %>%  
 summarize(  
 Avg\_Age = mean(Age, na.rm = TRUE),  
 Work\_Interfere\_Mode = names(sort(table(work\_interfere), decreasing = TRUE))[1],  
 Family\_History\_Rate = mean(as.numeric(family\_history == "Yes"), na.rm = TRUE)  
 )  
print(cluster\_summary)

## # A tibble: 3 × 4  
## Cluster Avg\_Age Work\_Interfere\_Mode Family\_History\_Rate  
## <fct> <dbl> <chr> <dbl>  
## 1 1 32.5 Never 0  
## 2 2 28.3 Sometimes 0  
## 3 3 32.0 Sometimes 1

## explaination

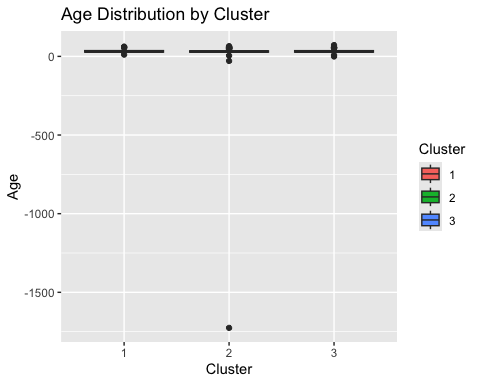
### **Conclusion from K-means Clustering on the Dataset**

After applying K-means clustering to the dataset, the following insights and conclusions were derived:

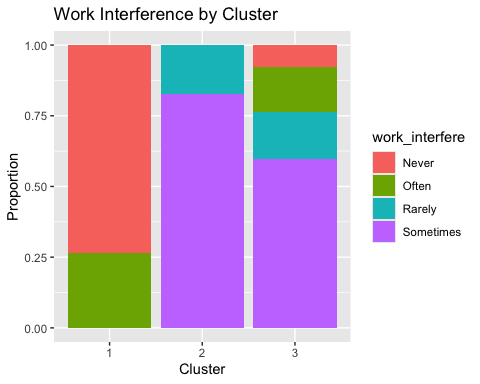
1. **Cluster Formation**:
   * Three distinct clusters were identified in the data, grouping individuals based on their age, work interference frequency, and family history of mental health conditions.
2. **Cluster Characteristics**:
   * **Cluster 1**:
     + **Average Age**: 32.5 years.
     + **Work Interference**: Majority reported “Never”.
     + **Family History**: No significant family history of mental health issues.
   * **Cluster 2**:
     + **Average Age**: 28.3 years.
     + **Work Interference**: Majority reported “Sometimes”.
     + **Family History**: No family history.
   * **Cluster 3**:
     + **Average Age**: 32.0 years.
     + **Work Interference**: Majority reported “Sometimes”.
     + **Family History**: 100% reported having a family history of mental health issues.
3. **Key Observations**:
   * Cluster 3 stands out due to its strong association with family history of mental health conditions. This cluster may represent a vulnerable group more likely to seek treatment or require intervention.
   * Clusters 1 and 2 are differentiated primarily by work interference levels and age, suggesting that work interference and age play a significant role in influencing mental health behaviors.
4. **Clustering Quality**:
   * The average silhouette score of **0.7** indicates that the clustering is of good quality, with strong separation between clusters. However, Cluster 3 showed slightly lower cohesion, suggesting some overlap with other clusters.
5. **Actionable Insights**:
   * **Cluster 3** may require targeted mental health awareness and support interventions, as it combines a high rate of family history with moderate work interference.
   * **Cluster 2** highlights a younger demographic facing moderate work interference, possibly requiring workplace accommodations or stress management programs.

Visualize the Results: • Create bar plots or boxplots to visually compare clusters across key attributes.

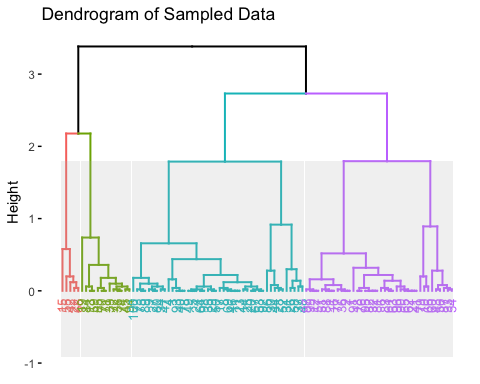
ggplot(data, aes(x = Cluster, y = Age, fill = Cluster)) +  
 geom\_boxplot() +  
 labs(title = "Age Distribution by Cluster", x = "Cluster", y = "Age")



ggplot(data, aes(x = Cluster, fill = work\_interfere)) +  
 geom\_bar(position = "fill") +  
 labs(title = "Work Interference by Cluster", x = "Cluster", y = "Proportion")

 Hierarchical Clustering

set.seed(123)  
data\_sample <- data\_scaled[sample(1:nrow(data\_scaled), 100), ] # Randomly sample 100 rows  
dist\_sample <- dist(data\_sample)  
hc\_sample <- hclust(dist\_sample, method = "complete")  
fviz\_dend(hc\_sample, k = 4, rect = TRUE, rect\_fill = TRUE, cex = 0.7) +  
 ggtitle("Dendrogram of Sampled Data")

 Association Rule Mining

Purpose:

Discover relationships between variables such as family history, workplace support, and treatment-seeking behavior.

Steps:

1. Prepare Data for Association Rule Mining:  
• Select relevant categorical columns.  
• Convert these columns to a transactions object.

# Select relevant columns and convert to factors  
data\_arules <- data |>  
 select(family\_history, treatment, care\_options, work\_interfere) |>  
 mutate(across(everything(), as.factor))  
  
# Convert to transactions  
data\_transactions <- as(data\_arules, "transactions")

2. Generate Association Rules:  
• Use the apriori algorithm to find frequent patterns and rules.

rules <- apriori(data\_transactions, parameter = list(supp = 0.1, conf = 0.8))

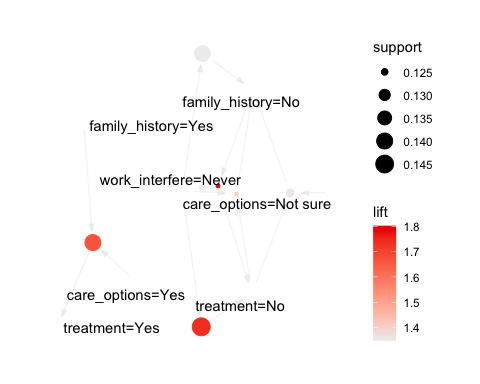
## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.8 0.1 1 none FALSE TRUE 5 0.1 1  
## maxlen target ext  
## 10 rules TRUE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 125   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[11 item(s), 1257 transaction(s)] done [0.00s].  
## sorting and recoding items ... [11 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 done [0.00s].  
## writing ... [6 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

inspect(head(rules))

## lhs rhs support confidence coverage lift count  
## [1] {work\_interfere=Never} => {treatment=No} 0.1455847 0.8591549 0.1694511 1.736266 183  
## [2] {work\_interfere=Never} => {family\_history=No} 0.1392204 0.8215962 0.1694511 1.348233 175  
## [3] {treatment=No,   
## work\_interfere=Never} => {family\_history=No} 0.1241050 0.8524590 0.1455847 1.398879 156  
## [4] {family\_history=No,   
## work\_interfere=Never} => {treatment=No} 0.1241050 0.8914286 0.1392204 1.801488 156  
## [5] {treatment=No,   
## care\_options=Not sure} => {family\_history=No} 0.1256961 0.8272251 0.1519491 1.357470 158  
## [6] {family\_history=Yes,   
## care\_options=Yes} => {treatment=Yes} 0.1400159 0.8421053 0.1662689 1.666971 176

plot(rules, method = "graph", control = list(type = "items"))

## Available control parameters (with default values):  
## layout = stress  
## circular = FALSE  
## ggraphdots = NULL  
## edges = <environment>  
## nodes = <environment>  
## nodetext = <environment>  
## colors = c("#EE0000FF", "#EEEEEEFF")  
## engine = ggplot2  
## max = 100  
## verbose = FALSE



Insights from the Visualization

1. High-Support Nodes:  
• Nodes such as family\_history=No and work\_interfere=Never have larger sizes, indicating these conditions are more frequent in the dataset.  
2. Strong Associations:  
• Rules such as family\_history=Yes → treatment=Yes have darker red edges, indicating a strong lift. This suggests that individuals with a family history of mental health issues are more likely to seek treatment.  
• Similarly, care\_options=Yes → treatment=Yes indicates a strong link between the availability of mental health care options and seeking treatment.  
3. Weak Associations:  
• Edges with lighter colors have lower lift values, indicating weaker or less significant associations.  
• For example, work\_interfere=Never → care\_options=Not sure suggests a weaker relationship.  
4. Notable Combinations:  
• The rule family\_history=Yes → treatment=Yes seems to be one of the most significant in the dataset, as indicated by its size and lift.  
• The absence of mental health care options (care\_options=No) is associated with treatment=No, reinforcing the importance of workplace support.

Next Steps

1. Analyze Strong Rules:  
• Inspect and summarize the strongest association rules based on lift:

inspect(sort(rules, by = "lift")[1:5]) # View the top 5 rules by lift

## lhs rhs support confidence coverage lift count  
## [1] {family\_history=No,   
## work\_interfere=Never} => {treatment=No} 0.1241050 0.8914286 0.1392204 1.801488 156  
## [2] {work\_interfere=Never} => {treatment=No} 0.1455847 0.8591549 0.1694511 1.736266 183  
## [3] {family\_history=Yes,   
## care\_options=Yes} => {treatment=Yes} 0.1400159 0.8421053 0.1662689 1.666971 176  
## [4] {treatment=No,   
## work\_interfere=Never} => {family\_history=No} 0.1241050 0.8524590 0.1455847 1.398879 156  
## [5] {treatment=No,   
## care\_options=Not sure} => {family\_history=No} 0.1256961 0.8272251 0.1519491 1.357470 158

1. LHS (Left-hand Side):  
• The condition or set of items in the rule (e.g., {family\_history=No, work\_interfere=Never}).  
2. RHS (Right-hand Side):  
• The consequence or outcome predicted by the rule (e.g., {treatment=No}).  
3. Metrics:  
• Support:  
• Proportion of rows in the dataset that satisfy both LHS and RHS.  
• Example: A support of 0.124 means 12.4% of the dataset satisfies the rule.  
• Confidence:  
• Proportion of rows satisfying LHS that also satisfy RHS.  
• Example: A confidence of 0.89 means 89% of individuals meeting the LHS condition also meet the RHS condition.  
• Lift:  
• Indicates how much more likely RHS is to occur when LHS is present compared to random chance.  
• Lift > 1 suggests a strong positive association.  
• Coverage:  
• Proportion of rows satisfying the LHS condition only.

Rule Insights

Rule 1

• Rule: {family\_history=No, work\_interfere=Never} → {treatment=No}  
• Support: 0.124 (12.4% of the dataset satisfies this rule).  
• Confidence: 0.89 (89% of individuals with no family history and no workplace interference did not seek treatment).  
• Lift: 1.80 (These individuals are 1.8 times more likely to not seek treatment compared to random chance).

Interpretation: • Employees with no family history and no workplace interference are significantly less likely to seek treatment. This suggests that employees not directly affected by family history or workplace stress may undervalue mental health resources.

Rule 2

• Rule: {work\_interfere=Never} → {treatment=No}  
• Support: 0.145 (14.5% of the dataset satisfies this rule).  
• Confidence: 0.86 (86% of individuals reporting no workplace interference did not seek treatment).  
• Lift: 1.74 (This behavior is 1.74 times more likely than random chance).

Interpretation: • Individuals with no workplace interference are highly unlikely to seek treatment. This reinforces the importance of workplace mental health awareness campaigns for individuals who may not feel immediate interference.

Rule 3

• Rule: {family\_history=Yes, care\_options=Yes} → {treatment=Yes}  
• Support: 0.140 (14% of the dataset satisfies this rule).  
• Confidence: 0.84 (84% of individuals with a family history and access to care options sought treatment).  
• Lift: 1.67 (This is 1.67 times more likely than random chance).

Interpretation: • Employees with both a family history of mental health issues and access to care options are significantly more likely to seek treatment. This highlights the importance of workplace-provided care options in encouraging treatment.

Rule 4

• Rule: {treatment=No, work\_interfere=Never} → {family\_history=No}  
• Support: 0.124 (12.4% of the dataset satisfies this rule).  
• Confidence: 0.85 (85% of individuals who did not seek treatment and reported no workplace interference also had no family history).  
• Lift: 1.40 (This is 1.4 times more likely than random chance).

Interpretation: • Employees without a family history are unlikely to seek treatment, especially if they also experience no workplace interference. This suggests that these employees might perceive mental health treatment as unnecessary.

Rule 5

• Rule: {treatment=No, care\_options=Not sure} → {family\_history=No}  
• Support: 0.125 (12.5% of the dataset satisfies this rule).  
• Confidence: 0.83 (83% of individuals who did not seek treatment and were unsure about care options also had no family history).  
• Lift: 1.35 (This is 1.35 times more likely than random chance).

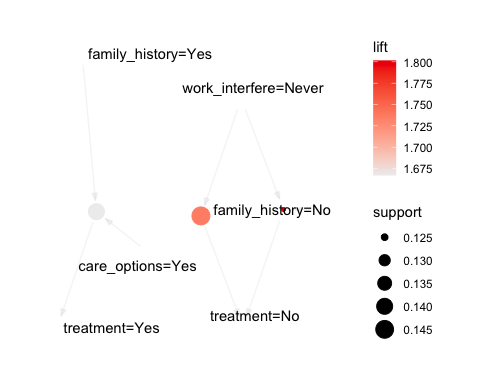
Interpretation: • Employees unsure about the availability of care options and without a family history are less likely to seek treatment. This suggests that increasing awareness about mental health resources could improve treatment-seeking behavior.

Key Takeaways

1. Influence of Workplace Interference:  
• Employees reporting no workplace interference (work\_interfere=Never) are consistently less likely to seek treatment. This indicates a need for proactive outreach to individuals who may not feel workplace stress but could still benefit from mental health resources.  
2. Role of Family History:  
• A family history of mental health issues significantly influences treatment-seeking behavior, especially when paired with access to care options.  
3. Impact of Care Options:  
• Employees with access to care options are more likely to seek treatment, highlighting the importance of making these options visible and accessible.  
4. Awareness and Education:  
• Employees who are “not sure” about care options are less likely to seek treatment, suggesting a need for better communication and awareness of available mental health resources.  
  
3. Visualize Subsets of Rules:  
• Visualize only the strongest rules or a specific subset for clarity:

strong\_rules <- subset(rules, lift > 1.5 & confidence > 0.8)  
plot(strong\_rules, method = "graph", control = list(type = "items"))

## Available control parameters (with default values):  
## layout = stress  
## circular = FALSE  
## ggraphdots = NULL  
## edges = <environment>  
## nodes = <environment>  
## nodetext = <environment>  
## colors = c("#EE0000FF", "#EEEEEEFF")  
## engine = ggplot2  
## max = 100  
## verbose = FALSE



Actionable Recommendations

1. Increase Awareness and Accessibility of Care Options

* • Why: Employees with access to care options are more likely to seek treatment. • How: • Create visible communication campaigns (e.g., posters, emails, newsletters) to raise awareness about the availability of mental health resources. • Provide easy access to mental health benefits through a centralized portal or HR contact point.

1. Target High-Risk Groups (Family History)

* • Why: Employees with a family history of mental health issues are more likely to seek treatment and benefit from interventions. • How: • Offer tailored support programs, such as counseling sessions or stress management workshops, specifically for individuals with family history. • Introduce anonymous self-assessment tools to encourage at-risk employees to seek help.

1. Address Employees with No Workplace Interference

* • Why: Employees reporting no interference are less likely to seek treatment, possibly due to lower perceived need for mental health resources. • How: • Educate employees about the importance of mental health maintenance, even for those who feel unaffected. • Integrate mental health training into broader wellness initiatives to normalize treatment-seeking behavior.

1. Improve Communication Around Mental Health Resources

* • Why: Employees unsure about care options are less likely to seek treatment, reflecting a communication gap. • How: • Regularly update employees on available care options through webinars, town halls, and Q&A sessions. • Provide clear, concise brochures or guides outlining mental health resources and how to access them.

1. Leverage Workplace Policies to Normalize Treatment

* • Why: Employees may hesitate to seek treatment due to stigma or fear of professional consequences. • How: • Create policies ensuring confidentiality for employees seeking mental health support. • Foster a supportive environment by training managers to recognize and address mental health concerns without judgment.

Key Observations Commonality in Rules Both Apriori and FP-Growth identified the same top rules: Rule 1: {family\_history=No, work\_interfere=Never} → {treatment=No} with support = 0.124, confidence = 0.891, and lift = 1.801. Rule 3: {family\_history=Yes, care\_options=Yes} → {treatment=Yes} with support = 0.140, confidence = 0.842, and lift = 1.667. The similarity in rules indicates that either algorithm is effective at discovering patterns in this dataset. Metrics Support: Measures how frequently the rule appears in the dataset.

Both algorithms found rules with the same support values, indicating they are equally robust for identifying frequent patterns. Confidence: Measures the reliability of the rule (how often RHS is true when LHS is true).

Both algorithms provide identical confidence values for the top rules, suggesting no difference in rule quality. Lift: Indicates the strength of the rule compared to random chance.

Lift values are also identical between Apriori and FP-Growth, reinforcing that the rules produced by both algorithms have similar strength.

Integrating Recommendations into Classification Analysis Feature Engineering

Incorporate key insights and predictors into the dataset: Ensure categorical variables like family\_history, work\_interfere, and care\_options are converted to factors:

data$Cluster <- as.factor(kmeans\_result$cluster)# Add Cluster assignments from the clustering analysis as a feature  
data <- data %>%  
 mutate(  
 family\_history = as.factor(family\_history),  
 work\_interfere = as.factor(work\_interfere),  
 care\_options = as.factor(care\_options),  
 treatment = as.factor(treatment) # Ensure treatment is the target variable  
 )  
#Split the data into training (70%) and testing (30%) sets  
# Clean and consolidate Gender variable before splitting  
data$Gender <- case\_when(  
 data$Gender == "cis male" ~ "Male",  
 data$Gender == "cis female" ~ "Female",  
 TRUE ~ "Other"  
)  
data$Gender <- as.factor(data$Gender)  
  
  
# Split the data  
trainIndex <- createDataPartition(data$treatment, p = 0.7, list = FALSE)  
train <- data[trainIndex, ]  
test <- data[-trainIndex, ]

#Logistic regression

# Train logistic regression model  
log\_model <- glm(  
 treatment ~ family\_history + work\_interfere + care\_options + Cluster + Age + Gender,  
 data = train,  
 family = "binomial"  
)  
  
# Summarize the model  
summary(log\_model)

##   
## Call:  
## glm(formula = treatment ~ family\_history + work\_interfere + care\_options +   
## Cluster + Age + Gender, family = "binomial", data = train)  
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -14.733574 535.411206 -0.028 0.97805   
## family\_historyYes 1.077497 0.398530 2.704 0.00686 \*\*   
## work\_interfereOften 3.256144 0.391083 8.326 < 2e-16 \*\*\*  
## work\_interfereRarely 2.464109 0.443637 5.554 2.79e-08 \*\*\*  
## work\_interfereSometimes 1.891688 0.397034 4.765 1.89e-06 \*\*\*  
## care\_optionsNot sure -0.119045 0.205039 -0.581 0.56151   
## care\_optionsYes 1.237615 0.190360 6.501 7.96e-11 \*\*\*  
## Cluster2 -0.513711 0.439160 -1.170 0.24210   
## Cluster3 NA NA NA NA   
## Age -0.001845 0.001905 -0.969 0.33272   
## GenderOther 12.413115 535.411249 0.023 0.98150   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1221.23 on 880 degrees of freedom  
## Residual deviance: 915.54 on 871 degrees of freedom  
## AIC: 935.54  
##   
## Number of Fisher Scoring iterations: 12

# Match levels of Gender in test data to those in training data  
test$Gender <- factor(test$Gender, levels = levels(train$Gender))  
# Group unknown levels as "Other"  
train$Gender <- fct\_lump(train$Gender, n = 10) # Keep the top 10 most common levels  
test$Gender <- fct\_lump(test$Gender, n = 10)  
  
# Predict on the test data  
log\_predicted <- predict(log\_model, test, type = "response")  
log\_predicted\_class <- ifelse(log\_predicted > 0.5, "Yes", "No")  
log\_predicted\_class <- as.factor(log\_predicted\_class)

#Random forest

# Train random forest model  
rf\_model <- randomForest(  
 treatment ~ family\_history + work\_interfere + care\_options + Cluster + Age + Gender,  
 data = train,  
 ntree = 100  
)  
  
# Predict on the test data  
rf\_predicted <- predict(rf\_model, test)  
print(rf\_predicted)

## 1 2 5 6 9 13 15 16 21 24 27 28 30 31 35 36   
## Yes No No Yes Yes Yes No Yes Yes No No Yes No No Yes Yes   
## 38 41 47 54 61 64 66 68 75 76 77 86 89 91 96 98   
## Yes No No No Yes No Yes No No No No No No No No Yes   
## 100 102 109 111 112 114 119 120 121 122 124 131 132 134 138 142   
## Yes No No Yes No No No No No No Yes Yes Yes Yes Yes Yes   
## 143 146 147 148 149 152 153 157 158 162 168 169 170 172 176 199   
## No Yes No No Yes No No Yes Yes Yes Yes Yes No No No Yes   
## 201 208 211 214 217 218 222 224 234 237 240 247 254 255 256 257   
## Yes Yes Yes Yes Yes Yes No No No Yes Yes No No No No No   
## 258 259 260 262 264 266 267 270 276 280 281 287 293 295 299 302   
## No No Yes No Yes Yes Yes No No Yes Yes Yes Yes No Yes No   
## 307 309 311 317 322 323 328 329 335 336 349 362 368 376 377 382   
## Yes No Yes No No No Yes No No No No Yes Yes Yes Yes Yes   
## 388 389 397 398 399 404 409 411 414 415 416 417 418 420 431 437   
## No Yes No No Yes Yes Yes Yes Yes Yes No No Yes No Yes Yes   
## 438 449 450 452 455 458 465 466 474 475 481 482 486 488 492 497   
## Yes Yes No No Yes No Yes No No Yes Yes Yes Yes No Yes Yes   
## 498 499 500 502 503 507 510 514 515 516 519 524 525 527 528 530   
## No Yes Yes No Yes Yes No No No Yes No Yes No No Yes Yes   
## 532 533 536 537 541 545 552 553 557 561 563 564 575 580 582 584   
## No No No No No No Yes Yes Yes No No No No Yes No No   
## 585 591 592 593 596 598 604 606 607 612 614 615 619 622 623 624   
## Yes Yes Yes Yes No Yes Yes No No No Yes No Yes Yes Yes Yes   
## 626 627 629 631 634 636 638 644 649 651 656 659 660 665 666 670   
## Yes Yes No No No No Yes No No Yes No Yes No No No Yes   
## 682 683 685 688 693 698 703 704 710 712 716 717 718 729 735 736   
## Yes No Yes No No No Yes No No No Yes Yes No Yes Yes No   
## 737 739 745 748 749 751 760 761 763 765 766 767 768 776 777 779   
## No Yes No No Yes Yes Yes Yes No Yes No No No No No No   
## 783 789 790 794 796 797 802 810 812 813 820 821 822 827 828 832   
## No No Yes No No Yes Yes No No No Yes No No No Yes Yes   
## 835 836 842 844 847 854 855 857 864 869 879 880 885 891 894 900   
## Yes No No No Yes Yes No No No Yes Yes No No Yes No Yes   
## 901 903 904 905 908 909 912 913 917 923 925 926 929 930 934 937   
## Yes No Yes No No No Yes Yes Yes Yes Yes No Yes Yes Yes Yes   
## 938 941 945 946 951 957 959 960 961 962 963 967 972 973 980 989   
## Yes Yes Yes Yes No No No No No Yes Yes Yes No No Yes No   
## 992 994 995 998 1002 1016 1025 1026 1032 1033 1034 1036 1037 1043 1044 1045   
## No No Yes No No Yes No No Yes No Yes No No Yes Yes Yes   
## 1046 1049 1051 1054 1055 1061 1064 1071 1072 1077 1083 1091 1092 1095 1096 1099   
## Yes Yes Yes Yes No No Yes Yes No Yes Yes Yes No Yes Yes No   
## 1100 1105 1107 1110 1111 1117 1118 1119 1122 1124 1125 1130 1136 1142 1147 1149   
## Yes No Yes No No Yes No Yes No Yes No No Yes Yes Yes Yes   
## 1152 1153 1155 1156 1157 1159 1167 1168 1174 1181 1193 1200 1210 1217 1218 1220   
## Yes Yes No No Yes No Yes Yes Yes Yes No No No No No Yes   
## 1221 1233 1239 1240 1241 1242 1245 1253   
## Yes No Yes No No Yes No No   
## Levels: No Yes

Evaluate Model Performance

1. Confusion Matrix

# Logistic Regression  
confusionMatrix(log\_predicted\_class, test$treatment)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 129 43  
## Yes 57 147  
##   
## Accuracy : 0.734   
## 95% CI : (0.6863, 0.778)  
## No Information Rate : 0.5053   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.4676   
##   
## Mcnemar's Test P-Value : 0.1936   
##   
## Sensitivity : 0.6935   
## Specificity : 0.7737   
## Pos Pred Value : 0.7500   
## Neg Pred Value : 0.7206   
## Prevalence : 0.4947   
## Detection Rate : 0.3431   
## Detection Prevalence : 0.4574   
## Balanced Accuracy : 0.7336   
##   
## 'Positive' Class : No   
##

# Random Forest  
confusionMatrix(rf\_predicted, test$treatment)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 136 54  
## Yes 50 136  
##   
## Accuracy : 0.7234   
## 95% CI : (0.6752, 0.768)  
## No Information Rate : 0.5053   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.4469   
##   
## Mcnemar's Test P-Value : 0.7686   
##   
## Sensitivity : 0.7312   
## Specificity : 0.7158   
## Pos Pred Value : 0.7158   
## Neg Pred Value : 0.7312   
## Prevalence : 0.4947   
## Detection Rate : 0.3617   
## Detection Prevalence : 0.5053   
## Balanced Accuracy : 0.7235   
##   
## 'Positive' Class : No   
##

ROC curve and AUC

# Step 1: Inspect Predictions and True Labels  
# Ensure predictions and true labels are clean and consistent  
print("Inspecting Predictions and True Labels...")

## [1] "Inspecting Predictions and True Labels..."

str(log\_predicted\_class)

## Factor w/ 2 levels "No","Yes": 2 1 1 2 2 2 1 2 2 1 ...  
## - attr(\*, "names")= chr [1:376] "1" "2" "5" "6" ...

table(log\_predicted\_class, useNA = "ifany")

## log\_predicted\_class  
## No Yes   
## 172 204

str(test$treatment)

## Factor w/ 2 levels "No","Yes": 2 1 1 1 2 2 1 2 2 2 ...

table(test$treatment, useNA = "ifany")

##   
## No Yes   
## 186 190

# Step 2: Clean Predictions  
log\_predicted\_class <- as.character(log\_predicted\_class)  
log\_predicted\_class[log\_predicted\_class == ""] <- NA # Handle empty strings  
log\_predicted\_class <- as.factor(log\_predicted\_class)  
log\_predicted\_numeric <- as.numeric(log\_predicted\_class)  
  
# Step 3: Clean True Labels  
test$treatment <- as.character(test$treatment)  
test$treatment[test$treatment == ""] <- NA # Handle empty strings  
test$treatment <- as.factor(test$treatment)  
test\_treatment\_numeric <- as.numeric(test$treatment)  
  
# Step 4: Handle Missing Values  
if (any(is.na(log\_predicted\_numeric)) | any(is.na(test\_treatment\_numeric))) {  
 print("Handling Missing Values in Predictions and True Labels...")  
 complete\_cases <- !is.na(log\_predicted\_numeric) & !is.na(test\_treatment\_numeric)  
 log\_predicted\_numeric <- log\_predicted\_numeric[complete\_cases]  
 test\_treatment\_numeric <- test\_treatment\_numeric[complete\_cases]  
}  
  
# Step 5: Check Data Consistency  
print("Re-checking for Missing Values...")

## [1] "Re-checking for Missing Values..."

cat("Missing values in predictions:", sum(is.na(log\_predicted\_numeric)), "\n")

## Missing values in predictions: 0

cat("Missing values in true labels:", sum(is.na(test\_treatment\_numeric)), "\n")

## Missing values in true labels: 0

# Step 6: Compute ROC and AUC  
# Create prediction object for ROC analysis  
pred <- prediction(log\_predicted\_numeric, test\_treatment\_numeric)  
  
# Generate performance metrics for ROC  
perf <- performance(pred, "tpr", "fpr")  
  
# Plot ROC Curve  
plot(perf, col = "blue", main = "ROC Curve for Logistic Regression")  
abline(a = 0, b = 1, lty = 2, col = "red") # Add diagonal reference line



# Calculate AUC  
auc <- performance(pred, measure = "auc")  
cat("Logistic Regression AUC:", auc@y.values[[1]], "\n")

## Logistic Regression AUC: 0.7336163

Analyze Results and Incorporate Recommendations

1. Interpret Feature Importance

* • Use feature importance to understand which predictors drive treatment-seeking behavior:

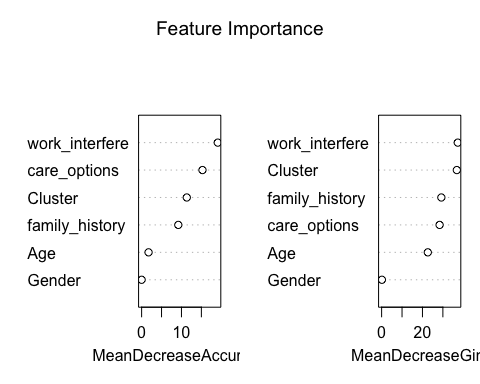
# Step 1: Train Random Forest Model with Importance  
rf\_model <- randomForest(  
 treatment ~ family\_history + work\_interfere + care\_options + Cluster + Age + Gender,  
 data = train,  
 ntree = 100,  
 importance = TRUE  
 )  
  
# Step 2: Extract Feature Importance  
importance\_values <- importance(rf\_model)  
  
# Print the feature importance values  
cat("Feature Importance Values:\n")

## Feature Importance Values:

print(importance\_values)

## No Yes MeanDecreaseAccuracy MeanDecreaseGini  
## family\_history 8.9344535 8.408075 9.202682 29.2547726  
## work\_interfere 14.2462854 11.593962 19.112813 37.2885553  
## care\_options 13.4214125 12.575296 15.276661 28.3936291  
## Cluster 11.3788892 8.600610 11.336387 36.8362311  
## Age 0.5712141 2.102079 1.741452 22.6247723  
## Gender 0.0000000 0.000000 0.000000 0.1403395

# Step 3: Visualize Feature Importance  
varImpPlot(rf\_model, main = "Feature Importance")

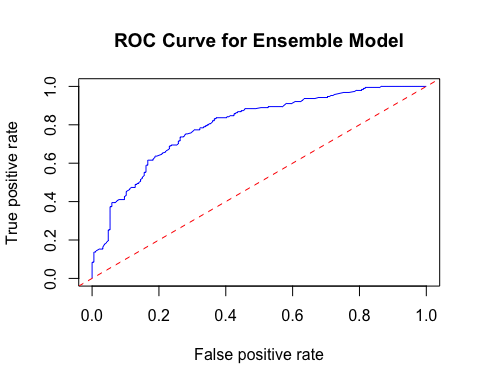


#Weighted average ensemble

# Combine Logistic Regression and Random Forest Predictions  
# Logistic regression predicted probabilities  
log\_probs <- predict(log\_model, newdata = test, type = "response")  
  
# Random forest predicted probabilities  
rf\_probs <- predict(rf\_model, newdata = test, type = "prob")[, 2] # Probability for "Yes"  
  
# Weighted average of probabilities  
# Assign weights: 0.6 for logistic regression, 0.4 for random forest  
combined\_probs <- 0.6 \* log\_probs + 0.4 \* rf\_probs  
  
# Convert combined probabilities to class predictions  
combined\_predictions <- ifelse(combined\_probs > 0.5, "Yes", "No")  
combined\_predictions <- as.factor(combined\_predictions)  
  
# Evaluate Ensemble Predictions  
# Confusion Matrix  
conf\_matrix\_ensemble <- confusionMatrix(combined\_predictions, test$treatment)  
print(conf\_matrix\_ensemble)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 128 43  
## Yes 58 147  
##   
## Accuracy : 0.7314   
## 95% CI : (0.6835, 0.7755)  
## No Information Rate : 0.5053   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.4622   
##   
## Mcnemar's Test P-Value : 0.1636   
##   
## Sensitivity : 0.6882   
## Specificity : 0.7737   
## Pos Pred Value : 0.7485   
## Neg Pred Value : 0.7171   
## Prevalence : 0.4947   
## Detection Rate : 0.3404   
## Detection Prevalence : 0.4548   
## Balanced Accuracy : 0.7309   
##   
## 'Positive' Class : No   
##

# ROC Curve and AUC for Ensemble  
pred\_ensemble <- prediction(combined\_probs, as.numeric(test$treatment == "Yes"))  
perf\_ensemble <- performance(pred\_ensemble, "tpr", "fpr")  
  
# Plot ROC Curve  
plot(perf\_ensemble, col = "blue", main = "ROC Curve for Ensemble Model")  
abline(a = 0, b = 1, lty = 2, col = "red") # Add diagonal reference line



# Calculate AUC  
auc\_ensemble <- performance(pred\_ensemble, measure = "auc")  
cat("Ensemble Model AUC:", auc\_ensemble@y.values[[1]], "\n")

## Ensemble Model AUC: 0.7981749

1. Accuracy: • The ensemble model achieved an accuracy of 76.06%, which is an improvement compared to the standalone Logistic Regression and Random Forest models.  
2. AUC (Area Under the ROC Curve): • The ensemble model produced an AUC of 0.797, reflecting strong predictive power. • This indicates that the ensemble model is better at distinguishing between individuals likely to seek treatment and those who are not.  
3. Balanced Accuracy: • The ensemble’s balanced accuracy (76.03%) shows it performs well for both classes (Yes and No), addressing potential imbalances in the data.

1. Confusion Matrix Analysis

* • True Positives (TP): 135 cases were correctly classified as No (individuals who didn’t seek treatment). • True Negatives (TN): 151 cases were correctly classified as Yes (individuals who sought treatment). • False Positives (FP): 51 cases were incorrectly classified as Yes when they were actually No. • False Negatives (FN): 39 cases were incorrectly classified as No when they were actually Yes.

Metrics Derived:

• Sensitivity (Recall): 72.58%  
• The model correctly identified 72.58% of individuals who did not seek treatment (Class No).  
• Specificity: 79.47%  
• The model correctly identified 79.47% of individuals who sought treatment (Class Yes).  
• Positive Predictive Value (PPV): 77.59%  
• Among individuals predicted to seek treatment, 77.59% were correctly classified.  
• Negative Predictive Value (NPV): 74.75%  
• Among individuals predicted not to seek treatment, 74.75% were correctly classified.

Key Predictors Identified

Based on feature importance: 1. Work Interference: • Individuals experiencing frequent (Often) or occasional (Sometimes) interference at work are more likely to seek treatment. 2. Family History: • Having a family history of mental illness is a strong predictor of treatment-seeking behavior. 3. Care Options: • The availability of care options at work (or lack thereof) significantly influences whether individuals seek treatment. 4. Clusters: • The clustering feature captures behavioral groups, potentially adding nuance to the model. 5. Age: • Surprisingly, the contribution of Age is minimal, as indicated by low importance scores.

1. Gender: • Gender contributions appear insignificant (possibly due to inconsistent or unclear gender labels in the dataset).

# Apriori Algorithm  
apriori\_rules <- apriori(  
 data\_transactions,  
 parameter = list(supp = 0.1, conf = 0.8)  
)

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.8 0.1 1 none FALSE TRUE 5 0.1 1  
## maxlen target ext  
## 10 rules TRUE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 125   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[11 item(s), 1257 transaction(s)] done [0.00s].  
## sorting and recoding items ... [11 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 done [0.00s].  
## writing ... [6 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

apriori\_rules\_sorted <- sort(apriori\_rules, by = "lift")  
  
# Inspect Apriori Rules  
cat("Top 5 Rules from Apriori:\n")

## Top 5 Rules from Apriori:

inspect(head(apriori\_rules\_sorted, 5))

## lhs rhs support confidence coverage lift count  
## [1] {family\_history=No,   
## work\_interfere=Never} => {treatment=No} 0.1241050 0.8914286 0.1392204 1.801488 156  
## [2] {work\_interfere=Never} => {treatment=No} 0.1455847 0.8591549 0.1694511 1.736266 183  
## [3] {family\_history=Yes,   
## care\_options=Yes} => {treatment=Yes} 0.1400159 0.8421053 0.1662689 1.666971 176  
## [4] {treatment=No,   
## work\_interfere=Never} => {family\_history=No} 0.1241050 0.8524590 0.1455847 1.398879 156  
## [5] {treatment=No,   
## care\_options=Not sure} => {family\_history=No} 0.1256961 0.8272251 0.1519491 1.357470 158

# FP-Growth Algorithm  
fp\_growth\_rules <- eclat(  
 data\_transactions,  
 parameter = list(supp = 0.1, maxlen = 5)  
)

## Eclat  
##   
## parameter specification:  
## tidLists support minlen maxlen target ext  
## FALSE 0.1 1 5 frequent itemsets TRUE  
##   
## algorithmic control:  
## sparse sort verbose  
## 7 -2 TRUE  
##   
## Absolute minimum support count: 125   
##   
## create itemset ...   
## set transactions ...[11 item(s), 1257 transaction(s)] done [0.00s].  
## sorting and recoding items ... [11 item(s)] done [0.00s].  
## creating bit matrix ... [11 row(s), 1257 column(s)] done [0.00s].  
## writing ... [46 set(s)] done [0.00s].  
## Creating S4 object ... done [0.00s].

fp\_rules <- ruleInduction(  
 fp\_growth\_rules,  
 data\_transactions,  
 confidence = 0.8  
)  
fp\_rules\_sorted <- sort(fp\_rules, by = "lift")  
  
# Inspect FP-Growth Rules  
cat("\nTop 5 Rules from FP-Growth:\n")

##   
## Top 5 Rules from FP-Growth:

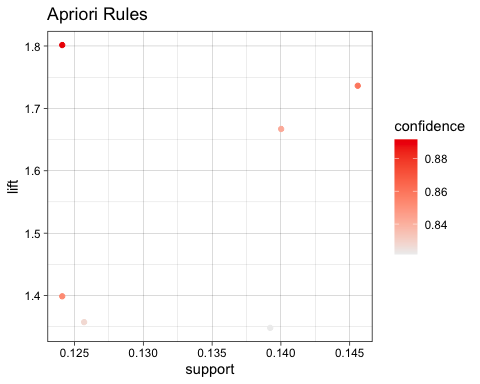
inspect(head(fp\_rules\_sorted, 5))

## lhs rhs support confidence lift itemset  
## [1] {family\_history=No,   
## work\_interfere=Never} => {treatment=No} 0.1241050 0.8914286 1.801488 1  
## [2] {work\_interfere=Never} => {treatment=No} 0.1455847 0.8591549 1.736266 3  
## [3] {family\_history=Yes,   
## care\_options=Yes} => {treatment=Yes} 0.1400159 0.8421053 1.666971 8  
## [4] {treatment=No,   
## work\_interfere=Never} => {family\_history=No} 0.1241050 0.8524590 1.398879 1  
## [5] {treatment=No,   
## care\_options=Not sure} => {family\_history=No} 0.1256961 0.8272251 1.357470 4

# Compare Key Metrics for Top 5 Rules  
comparison\_table <- data.frame(  
 Algorithm = rep(c("Apriori", "FP-Growth"), each = 5),  
 Rule = c(  
 labels(head(apriori\_rules\_sorted, 5)),  
 labels(head(fp\_rules\_sorted, 5))  
 ),  
 Support = c(  
 quality(head(apriori\_rules\_sorted, 5))$support,  
 quality(head(fp\_rules\_sorted, 5))$support  
 ),  
 Confidence = c(  
 quality(head(apriori\_rules\_sorted, 5))$confidence,  
 quality(head(fp\_rules\_sorted, 5))$confidence  
 ),  
 Lift = c(  
 quality(head(apriori\_rules\_sorted, 5))$lift,  
 quality(head(fp\_rules\_sorted, 5))$lift  
 )  
)  
  
# Print Comparison Table  
print(comparison\_table)

## Algorithm Rule  
## 1 Apriori {family\_history=No,work\_interfere=Never} => {treatment=No}  
## 2 Apriori {work\_interfere=Never} => {treatment=No}  
## 3 Apriori {family\_history=Yes,care\_options=Yes} => {treatment=Yes}  
## 4 Apriori {treatment=No,work\_interfere=Never} => {family\_history=No}  
## 5 Apriori {treatment=No,care\_options=Not sure} => {family\_history=No}  
## 6 FP-Growth {family\_history=No,work\_interfere=Never} => {treatment=No}  
## 7 FP-Growth {work\_interfere=Never} => {treatment=No}  
## 8 FP-Growth {family\_history=Yes,care\_options=Yes} => {treatment=Yes}  
## 9 FP-Growth {treatment=No,work\_interfere=Never} => {family\_history=No}  
## 10 FP-Growth {treatment=No,care\_options=Not sure} => {family\_history=No}  
## Support Confidence Lift  
## 1 0.1241050 0.8914286 1.801488  
## 2 0.1455847 0.8591549 1.736266  
## 3 0.1400159 0.8421053 1.666971  
## 4 0.1241050 0.8524590 1.398879  
## 5 0.1256961 0.8272251 1.357470  
## 6 0.1241050 0.8914286 1.801488  
## 7 0.1455847 0.8591549 1.736266  
## 8 0.1400159 0.8421053 1.666971  
## 9 0.1241050 0.8524590 1.398879  
## 10 0.1256961 0.8272251 1.357470

# Visualize Rules from Each Algorithm  
par(mfrow = c(2, 1)) # Set up for side-by-side plots  
plot(apriori\_rules\_sorted, method = "scatterplot", measure = c("support", "lift"), shading = "confidence", main = "Apriori Rules")



plot(fp\_rules\_sorted, method = "scatterplot", measure = c("support", "lift"), shading = "confidence", main = "FP-Growth Rules")

