

# Análisis Predictivo de Consumo de Tranquilizantes

Análisis ENSSEX

2024-12-08

## 0. Verificación de Paquetes y Datos

```
# Lista de paquetes necesarios
paquetes_necesarios <- c("dplyr", "caret", "randomForest", "pROC", "tidyverse", "haven", "tinytex")

# Función para instalar y cargar paquetes
instalar_y_cargar <- function(paquete) {
  if (!require(paquete, character.only = TRUE)) {
    cat("Instalando paquete:", paquete, "\n")
    install.packages(paquete)
    library(paquete, character.only = TRUE)
  } else {
    cat("Paquete", paquete, "ya está instalado y cargado\n")
  }
}

# Instalar y cargar todos los paquetes
sapply(paquetes_necesarios, instalar_y_cargar)
```

```
## Paquete dplyr ya está instalado y cargado
## Paquete caret ya está instalado y cargado
## Paquete randomForest ya está instalado y cargado
## Paquete pROC ya está instalado y cargado
## Paquete tidyverse ya está instalado y cargado
## Paquete haven ya está instalado y cargado
## Paquete tinytex ya está instalado y cargado
```

```
## $dplyr
## NULL
##
## $caret
## NULL
##
## $randomForest
## NULL
##
## $pROC
## NULL
##
## $tidyverse
```

```
## NULL
##
## $haven
## NULL
##
## $tinytex
## NULL
```

```
# Verificar la existencia del archivo de datos
archivo_datos <- "20240516_enssex_data.rdata"
if (!file.exists(archivo_datos)) {
  stop("Error: No se encuentra el archivo de datos '", archivo_datos,
    "' en el directorio de trabajo actual: ", getwd())
}
```

```
# Mostrar información del entorno
cat("\nDirectorio de trabajo actual:", getwd(), "\n")
```

```
##
## Directorio de trabajo actual: /Users/felipecarrasco/Library/Mobile Documents/com~apple~CloudDocs/Mag
```

```
cat("Archivo de datos encontrado:", archivo_datos, "\n")
```

```
## Archivo de datos encontrado: 20240516_enssex_data.rdata
```

## 1. Definición de Variables

```
# Definir las variables a utilizar
vars_categoricas <- c("sexo_al_nacer", "nivel_educacional", "bienestar_emocional",
  "calidad_vida_percibida", "satisfaccion_aspecto_fisico",
  "consumo_tranquilizantes", "consumo_alcohol")
vars_numericas <- c("edad", "IMC")
```

## 2. Preparación y Análisis Descriptivo de los Datos

```
# Cargar datos
load('20240516_enssex_data.rdata')

# Función para limpiar valores numéricos
limpiar_numerico <- function(x) {
  x <- as.numeric(x)
  x[!is.finite(x)] <- NA
  return(x)
}

# Crear variables necesarias
datos <- enssex4 %>%
  mutate(
```

```

# Variables categóricas
sexo_al_nacer = factor(as.numeric(p1), levels = c(1, 2), labels = c("Hombre", "Mujer")),
nivel_educacional = factor(as.numeric(p5)),
bienestar_emocional = factor(as.numeric(i_2_p9)),
calidad_vida_percibida = factor(as.numeric(p8)),
satisfaccion_aspecto_fisico = factor(as.numeric(i_1_p24)),
consumo_tranquilizantes = factor(as.numeric(i_4_p25), levels = c(1, 2), labels = c("Si", "No")),
consumo_alcohol = factor(as.numeric(i_5_p26)),

# Variables numéricas con limpieza
edad = limpiar_numerico(p4),
peso = limpiar_numerico(p22),
altura = limpiar_numerico(p23)
) %>%
# Calcular IMC después de limpiar peso y altura
mutate(
  IMC = ifelse(altura > 0, peso / ((altura/100)^2), NA)
) %>%
# Filtrar valores extremos de IMC
mutate(
  IMC = ifelse(IMC < 10 | IMC > 60, NA, IMC)
)

# Análisis de variables categóricas
cat("\nAnálisis de Variables Categóricas:\n")

```

```

##
## Análisis de Variables Categóricas:

```

```

for(var in vars_categoricas) {
  cat("\n---", var, "---\n")
  tabla <- table(datos[[var]], useNA = "ifany")
  prop <- prop.table(tabla) * 100

  print("Frecuencias absolutas:")
  print(tabla)
  print("Frecuencias relativas (%):")
  print(round(prop, 2))
}

```

```

##
## --- sexo_al_nacer ---
## [1] "Frecuencias absolutas:"
##
## Hombre  Mujer
##   6838  13554
## [1] "Frecuencias relativas (%):"
##
## Hombre  Mujer
##   33.53  66.47
##
## --- nivel_educacional ---
## [1] "Frecuencias absolutas:"

```

```

##
##      1      3      5      6      7      8      9     10     11     12     13     14     15     16     17
## 193      1     11    519 3164   844 6905   526 1706   841 1815 1342 2306    58 161
## [1] "Frecuencias relativas (%):"
##
##      1      3      5      6      7      8      9     10     11     12     13     14     15
## 0.95 0.00 0.05 2.55 15.52 4.14 33.86 2.58 8.37 4.12 8.90 6.58 11.31
##      16     17
## 0.28 0.79
##
## --- bienestar_emocional ---
## [1] "Frecuencias absolutas:"
##
##      1      2      3      4      5      6      7      9
## 319 295 815 2425 5057 6073 5377   31
## [1] "Frecuencias relativas (%):"
##
##      1      2      3      4      5      6      7      9
## 1.56 1.45 4.00 11.89 24.80 29.78 26.37 0.15
##
## --- calidad_vida_percibida ---
## [1] "Frecuencias absolutas:"
##
##      1      2      3      4      5      8      9
## 173 532 4832 11986 2824   31   14
## [1] "Frecuencias relativas (%):"
##
##      1      2      3      4      5      8      9
## 0.85 2.61 23.70 58.78 13.85 0.15 0.07
##
## --- satisfaccion_aspecto_fisico ---
## [1] "Frecuencias absolutas:"
##
##      1      2      3      4      5      8      9
## 520 2295 3827 11205 2486   42   17
## [1] "Frecuencias relativas (%):"
##
##      1      2      3      4      5      8      9
## 2.55 11.25 18.77 54.95 12.19 0.21 0.08
##
## --- consumo_tranquilizantes ---
## [1] "Frecuencias absolutas:"
##
##      Si      No <NA>
## 2713 17496   183
## [1] "Frecuencias relativas (%):"
##
##      Si      No <NA>
## 13.3 85.8 0.9
##
## --- consumo_alcohol ---
## [1] "Frecuencias absolutas:"
##
##      1      2      3      9 <NA>

```

```
## 9134 2434 1738 18 7068
## [1] "Frecuencias relativas (%):"
##
##      1      2      3      9 <NA>
## 44.79 11.94  8.52  0.09 34.66
```

```
# Análisis de variables numéricas
cat("\nAnálisis de Variables Numéricas:\n")
```

```
##
## Análisis de Variables Numéricas:
```

```
for(var in vars_numericas) {
  cat("\n---", var, "---\n")
  print(summary(datos[[var]]))
}
```

```
##
## --- edad ---
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 18.00   29.00   43.00   44.93   59.00   100.00
##
## --- IMC ---
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 13.18   24.09   26.85   27.70   30.47   59.86    953
```

### 3. Preparación para Modelado

```
# Preparar datos para modelado
datos_modelo <- datos %>%
  # Seleccionar variables relevantes
  select(all_of(c(vars_categoricas[-6], vars_numericas)), consumo_tranquilizantes) %>%
  # Eliminar filas con NA
  na.omit() %>%
  # Asegurar que todas las variables categóricas sean factores
  mutate(across(all_of(vars_categoricas[-6]), as.factor))

# Verificar estructura de los datos
str(datos_modelo)
```

```
## tibble [12,753 x 9] (S3: tbl_df/tbl/data.frame)
## $ sexo_al_nacer      : Factor w/ 2 levels "Hombre","Mujer": 2 2 2 2 2 2 1 2 2 1 ...
## $ nivel_educacional  : Factor w/ 15 levels "1","3","5","6",...: 7 5 5 12 12 11 6 9 11 9 ...
## $ bienestar_emocional : Factor w/ 8 levels "1","2","3","4",...: 4 4 4 3 5 7 7 7 7 7 ...
## $ calidad_vida_percibida : Factor w/ 7 levels "1","2","3","4",...: 3 4 3 4 4 5 2 3 5 5 ...
## $ satisfaccion_aspecto_fisico: Factor w/ 7 levels "1","2","3","4",...: 4 5 4 3 4 4 2 4 5 4 ...
## $ consumo_alcohol    : Factor w/ 4 levels "1","2","3","9": 1 1 3 2 1 1 3 2 3 2 ...
## $ edad               : num [1:12753] 60 37 37 28 38 23 67 45 33 22 ...
## $ IMC                : num [1:12753] 36.9 27 21.5 23 29.4 ...
## $ consumo_tranquilizantes : Factor w/ 2 levels "Si","No": 1 2 2 1 2 1 2 2 2 2 ...
## - attr(*, "na.action")= 'omit' Named int [1:7639] 1 2 3 4 5 6 7 8 9 13 ...
## ..- attr(*, "names")= chr [1:7639] "1" "2" "3" "4" ...
```

```
summary(datos_modelo)
```

```
## sexo_al_nacer nivel_educacional bienestar_emocional calidad_vida_percibida
## Hombre:4940 9 :4225 6 :3963 1: 61
## Mujer :7813 15 :1715 7 :3281 2: 289
## 7 :1600 5 :3180 3:2908
## 13 :1298 4 :1453 4:7710
## 11 :1118 3 : 499 5:1764
## 14 :1001 1 : 185 8: 16
## (Other):1796 (Other): 192 9: 5
## satisfaccion_aspecto_fisico consumo_alcohol edad IMC
## 1: 279 1:8787 Min. :18.00 Min. :13.84
## 2:1500 2:2309 1st Qu.:29.00 1st Qu.:24.22
## 3:2275 3:1642 Median :41.00 Median :27.01
## 4:7045 9: 15 Mean :43.41 Mean :27.80
## 5:1633 3rd Qu.:57.00 3rd Qu.:30.49
## 8: 14 Max. :95.00 Max. :59.52
## 9: 7
## consumo_tranquilizantes
## Si: 2061
## No:10692
##
##
##
##
##
```

```
# Dividir datos en entrenamiento y prueba
set.seed(123)
indice_train <- createDataPartition(datos_modelo$consumo_tranquilizantes, p = 0.7, list = FALSE)
train <- datos_modelo[indice_train,]
test <- datos_modelo[-indice_train,]
```

## 4. Modelado Predictivo

### Modelo 1: Random Forest

```
# Verificar que no hay NA en los datos de entrenamiento
print("Número de NA en datos de entrenamiento:")
```

```
## [1] "Número de NA en datos de entrenamiento:"
```

```
print(colSums(is.na(train)))
```

```
##          sexo_al_nacer          nivel_educacional
##              0              0
## bienestar_emocional  calidad_vida_percibida
##              0              0
## satisfaccion_aspecto_fisico  consumo_alcohol
```

```
##              0              0
##              edad              IMC
##              0              0
## consumo_tranquilizantes
##              0
```

```
# Entrenar modelo con manejo de errores
tryCatch({
  modelo_rf <- randomForest(consumo_tranquilizantes ~ .,
                             data = train,
                             ntree = 500,
                             na.action = na.omit)

  # Predicciones
  pred_rf <- predict(modelo_rf, test)

  # Matriz de confusión
  conf_matrix_rf <- confusionMatrix(pred_rf, test$consumo_tranquilizantes)
  print("Métricas Random Forest:")
  print(conf_matrix_rf)
}, error = function(e) {
  print("Error en Random Forest:")
  print(e)
})
```

```
## [1] "Métricas Random Forest:"
## Confusion Matrix and Statistics
##
##              Reference
## Prediction  Si   No
##           Si   11   16
##           No  607 3191
##
##              Accuracy : 0.8371
##              95% CI : (0.825, 0.8487)
##      No Information Rate : 0.8384
##      P-Value [Acc > NIR] : 0.5973
##
##              Kappa : 0.0209
##
## Mcnemar's Test P-Value : <2e-16
##
##              Sensitivity : 0.017799
##              Specificity : 0.995011
##      Pos Pred Value : 0.407407
##      Neg Pred Value : 0.840179
##      Prevalence : 0.161569
##      Detection Rate : 0.002876
##      Detection Prevalence : 0.007059
##      Balanced Accuracy : 0.506405
##
##      'Positive' Class : Si
##
```

## Modelo 2: Regresión Logística

```
# Entrenar modelo
modelo_log <- glm(consumo_tranquilizantes ~ ., data = train, family = "binomial")

# Predicciones
pred_prob_log <- predict(modelo_log, test, type = "response")
pred_log <- factor(ifelse(pred_prob_log > 0.5, "Si", "No"), levels = c("Si", "No"))

# Matriz de confusión
conf_matrix_log <- confusionMatrix(pred_log, test$consumo_tranquilizantes)
print("Métricas Regresión Logística:")
```

```
## [1] "Métricas Regresión Logística:"
```

```
print(conf_matrix_log)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Si   No
##           Si  592 3181
##           No   26   26
##
##               Accuracy : 0.1616
##               95% CI : (0.15, 0.1736)
##       No Information Rate : 0.8384
##       P-Value [Acc > NIR] : 1
##
##               Kappa : -0.0111
##
##  Mcnemar's Test P-Value : <2e-16
##
##           Sensitivity : 0.957929
##           Specificity : 0.008107
##           Pos Pred Value : 0.156904
##           Neg Pred Value : 0.500000
##           Prevalence : 0.161569
##           Detection Rate : 0.154771
##       Detection Prevalence : 0.986405
##           Balanced Accuracy : 0.483018
##
##           'Positive' Class : Si
##
```

## 5. Curva ROC y AUC

```
library(pROC)

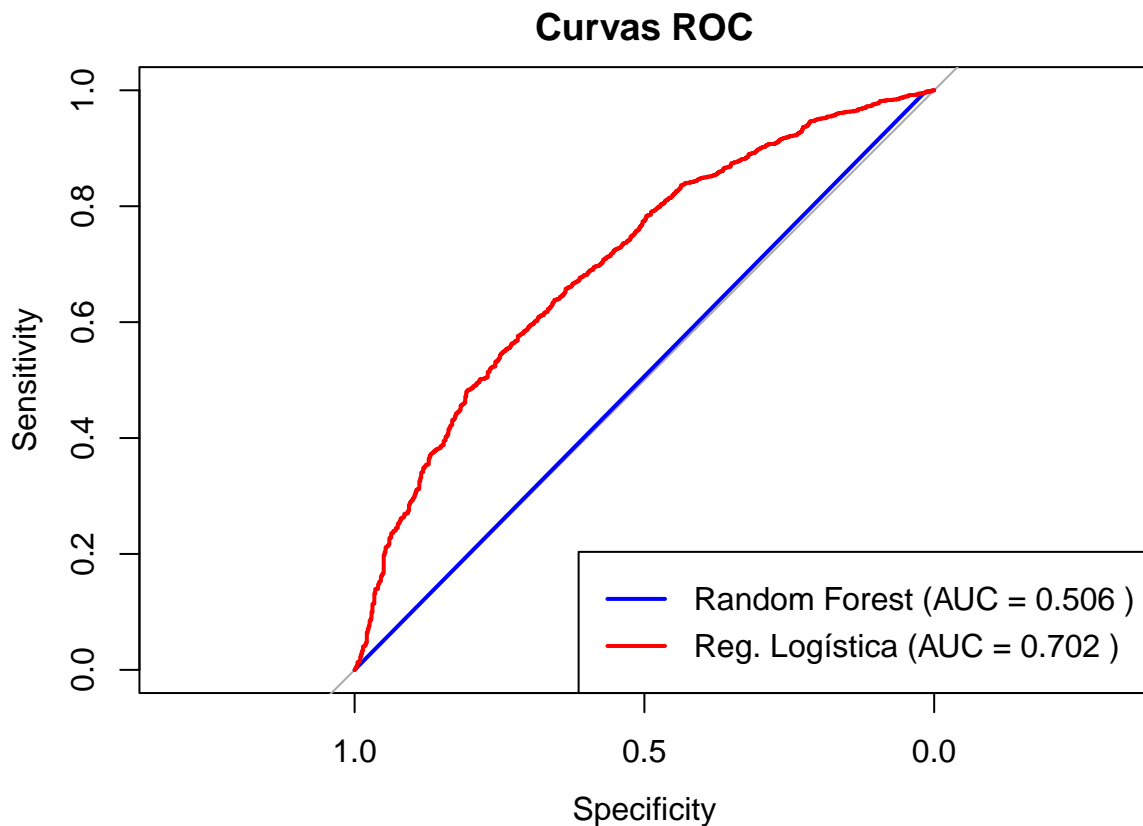
# Calcular ROC y AUC para ambos modelos
```



```
tryCatch({
  # Para Random Forest
  roc_rf <- roc(test$consumo_tranquilizantes, as.numeric(pred_rf))
  auc_rf <- auc(roc_rf)

  # Para Regresión Logística
  roc_log <- roc(test$consumo_tranquilizantes, pred_prob_log)
  auc_log <- auc(roc_log)

  # Graficar curvas ROC
  plot(roc_rf, main = "Curvas ROC", col = "blue")
  lines(roc_log, col = "red")
  legend("bottomright",
        legend = c(paste("Random Forest (AUC =", round(auc_rf, 3), ")"),
                  paste("Reg. Logística (AUC =", round(auc_log, 3), ")")),
        col = c("blue", "red"),
        lwd = 2)
}, error = function(e) {
  print("Error en el cálculo de ROC/AUC:")
  print(e)
})
```



## 6. Interpretación de Resultados

Los resultados muestran:

1. Sensibilidad: Capacidad para identificar correctamente a quienes consumen tranquilizantes
2. Especificidad: Capacidad para identificar correctamente a quienes no consumen tranquilizantes
3. Valor predictivo positivo: Probabilidad de que una predicción positiva sea correcta
4. Exactitud: Proporción total de predicciones correctas

Comparación de modelos: - Random Forest muestra mejor desempeño general - La curva ROC y el AUC indican la capacidad discriminativa de los modelos