Modelo de Pesquisa Automatizada del Riesgo Cardiovascular mediante Inteligencia Artificial en la Red APS de Quellón, provincia de Chiloé

Claudio Cárdenas M.

```
## Carga de paquetes de analisis

## update.packages(ask = FALSE)  ##Actualizar todos los paquetes desde CRAN

if (!requireNamespace("pacman", quietly = TRUE)) {
  install.packages("pacman")
}

pacman::p_load(dplyr,
               tinytex,
               latexpdf,
               readxl,
               purrr,
               ggplot2,
               skimr,
               GGally,
               tidyr,
               mice,
               patchwork,
               rstatix,
               writexl,
               caret,
               rlang,
               fastmap,
               ROCR,
               MASS,
               gridExtra,
               pROC,
               gbm,
               xgboost,
               funModeling,
               broom)
```

1 ETAPA 1: Comprensión del negocio.

1.1 Identificación de los objetivos del negocio y situación actual.

La comuna de Quellón, situada en la provincia de Chiloé, enfrenta desafíos significativos en salud pública debido a la alta prevalencia de factores de riesgo cardiovascular (FRCV) en su población. Estudios realizados

en diversas regiones de Chile han evidenciado tasas elevadas de hipertensión arterial, dislipidemias, obesidad y diabetes mellitus tipo 2 . Aunque no se dispone de datos específicos para Quellón, es razonable inferir que estas condiciones también afectan a su población, considerando las similitudes sociodemográficas y epidemiológicas con otras zonas del país.

1.2 Objetivos del proyecto de *Data Science.*

Entrenar un modelo predictivo basado en machine learning en la Atención Primaria de Salud (APS) de Quellón permitiría:

- **Mejorar la focalización de intervenciones preventivas**: Al identificar a individuos con alto riesgo cardiovascular, se pueden dirigir recursos y programas de prevención de manera más eficiente.
- **Priorizar controles de salud cardiovascular**: Facilitando la programación de controles y seguimientos para aquellos pacientes con mayor probabilidad de desarrollar enfermedades cardiovasculares.
- **Reducir eventos coronarios mayores y hospitalizaciones evitables**: La detección temprana y la intervención oportuna pueden disminuir la incidencia de eventos adversos graves .
- **Optimizar el uso de recursos en la red APS local**: Permitiendo una asignación más efectiva de los recursos disponibles, mejorando la eficiencia del sistema de salud.

La aplicación de algoritmos de machine learning en el ámbito de la salud ha demostrado ser eficaz en la predicción de riesgos y en la toma de decisiones clínicas . Estos modelos pueden analizar grandes volúmenes de datos y detectar patrones complejos que podrían pasar desapercibidos mediante métodos tradicionales.

En el contexto de Quellón, la adopción de esta tecnología podría representar un avance significativo en la gestión de la salud pública, contribuyendo a la reducción de la carga de enfermedades cardiovasculares y mejorando la calidad de vida de sus habitantes.

1.3 Planificación del proyecto de *Data Science.*

1.3.1 Objetivo general del estudio:

Entrenar un modelo predictivo basado en algoritmos de machine learning para la identificación temprana de personas en riesgo cardiovascular, utilizando datos clínicos de la población usuaria de la Atención Primaria de Salud en la comuna de Quellón, con el propósito de optimizar la focalización de intervenciones preventivas y mejorar la gestión del riesgo en salud cardiovascular.

1.3.2 Algoritmo de clasificación para el aprendizaje automático en el estudio.

La siguiente figura es un mapa conceptual de los principales algoritmos de clasificación supervisada, partiendo de un nodo central (“Algoritmos de Clasificación Supervisada”) y desplegando ramas coloreadas para cada técnica. Cada rama está estructurada en tres secciones:

- Definición (qué criterio o función utiliza el algoritmo para clasificar),
- Ventajas (fortalezas en términos de interpretabilidad, velocidad o robustez),
- Desventajas (limitaciones relacionadas con supuestos, sensibilidad a hiperparámetros, coste computacional o infraestructuras de datos).

Las técnicas incluidas abarcan desde modelos lineales simples (Regresión Logística, Lasso/Ridge) hasta métodos de ensamble (Random Forest, Boosting) y enfoques no lineales o basados en vecinos (SVM, k-NN), así como Redes Neuronales y Naive Bayes. Este esquema ofrece una visión de conjunto que permite, de modo ágil y comparativo, seleccionar la estrategia más adecuada para un proyecto de modelación predictiva—por ejemplo, en la estimación de riesgo cardiovascular—según el volumen de datos, la necesidad de explicación clínica y los recursos computacionales disponibles.

1.3.3 Metodologia para el proceso de *Data Science*.

Para el desarrollo de este proyecto se ha empleado, y se empleará, la metodología CRISP-DM, la cual es de libre distribución y compatible con cualquier conjunto de herramientas de minería de datos. Dicha metodología estructura el ciclo de vida de un proyecto de minería de datos en seis fases interactivas e iterativas:

- Comprensión del negocio: definición de los objetivos, evaluación de la situación actual y elaboración del plan de trabajo.
- Comprensión de los datos: identificación y recopilación de las fuentes de datos disponibles.
- Preparación de los datos: limpieza, transformación e integración de los datos para su análisis.
- Modelado: aplicación de técnicas de análisis y generación de modelos que respondan a los objetivos del negocio, produciendo información nueva y relevante.
- Evaluación: análisis de los resultados obtenidos, valoración de la calidad y validación de los modelos desarrollados.
- Despliegue: planificación de la implementación y distribución de los resultados.

Las fases de evaluación y despliegue no se abordaran en el presente estudio, dado que excede el alcance del Magíster por las limitaciones de tiempo, recursos humanos y presupuesto. No obstante, se prevé su ejecución eventual en el caso de que sea necesario implementar en producción el modelo predictivo que demuestre mejor rendimiento en este trabajo.

1.3.4 Estructura de Desglose del Trabajo (EDT).

En la estructura de desglose de trabajo para el Proyecto Data Science: Modelo Predictivo de Pesquisa Cardiovascular, las etapas 1 a 4 — resaltadas en verde — constituyen el alcance operativo de este estudio y serán abordadas en su totalidad:

- Etapa 1. Comprensión del negocio Definición de objetivos clínicos y administrativos en la Red de Atención Primaria, evaluación del contexto institucional y planificación del proyecto.
- Etapa 2. Comprensión de los datos Identificación, recolección y caracterización de las fuentes de información epidemiológica y de salud disponibles.
- Etapa 3. Preparación de los datos Limpieza, transformación y estructuración de los registros para su uso en técnicas analíticas.
- Etapa 4. Modelación Selección de algoritmos, generación de diseños de experimentación y construcción de modelos predictivos orientados a la detección temprana de riesgo cardiovascular.

Por el contrario, las etapas 5 y 6 — señaladas en rojo — quedan fuera del presente estudio:

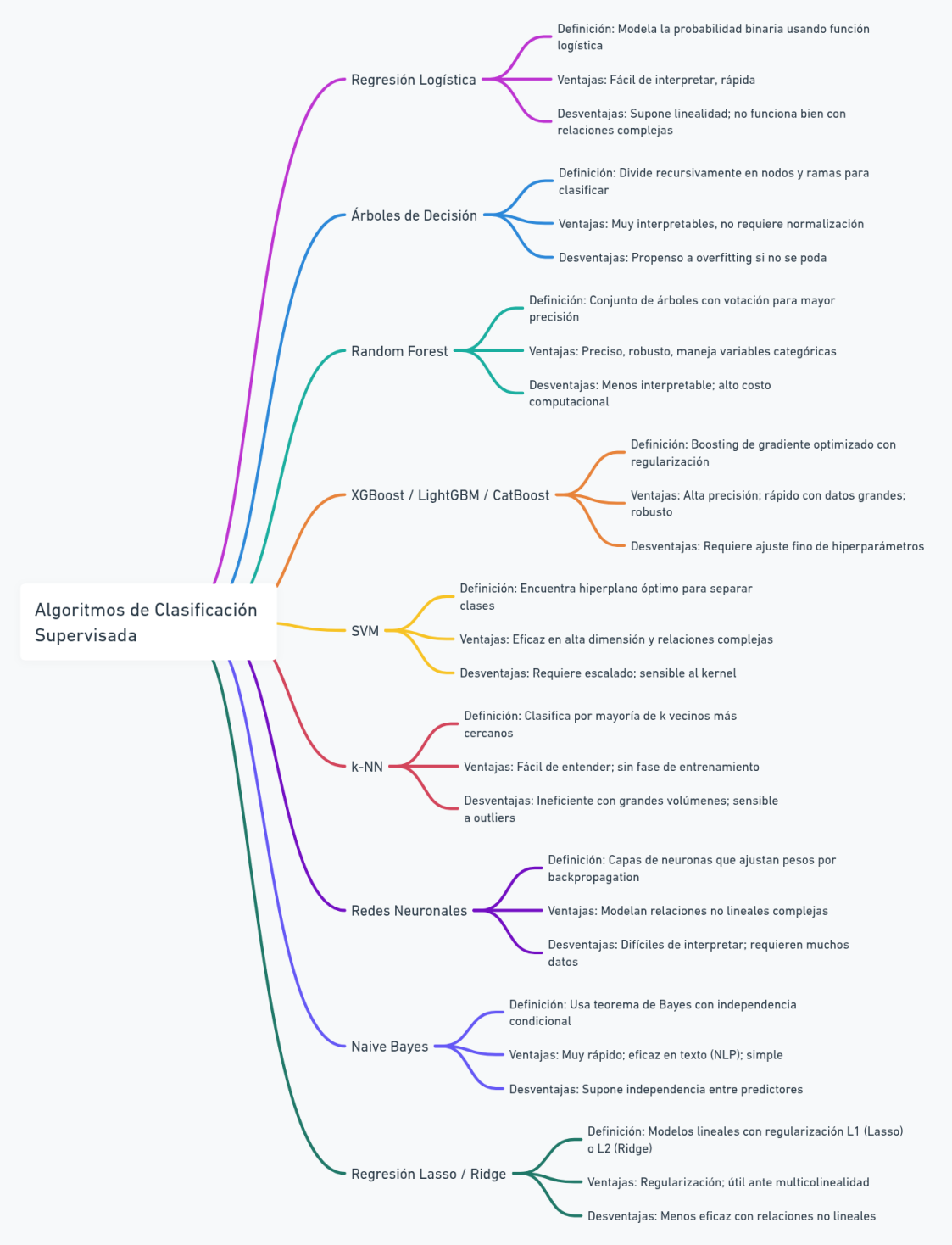


Figure 1: Comparación de algoritmos de Machine Learning, para clasificación supervisada (Fuente: <https://doi.org/10.1186/s12911-019-1004-8>)

Figure 2: Metodología CRISP-DM (Cross Industry Standard Process for Data Mining) <https://healthdataminer.com/data-mining/crisp-dm-una-metodologia-para-mineria-de-datos-en-salud/>

- Etapa 5. Evaluación avanzada Revisión en profundidad de calidad, validación externa y determinación de pasos de escalamiento.
- Etapa 6. Despliegue operativo Planificación de la implementación en entornos hospitalarios o de APS, definición de protocolos de mantenimiento y elaboración de informes para su integración en la gestión asistencial.

La exclusión de estas últimas fases responde a las limitaciones de tiempo y recursos propias del programa de Magíster, así como al hecho de que su ejecución corresponde a la etapa de transición a producción de un modelo en un entorno real de salud.

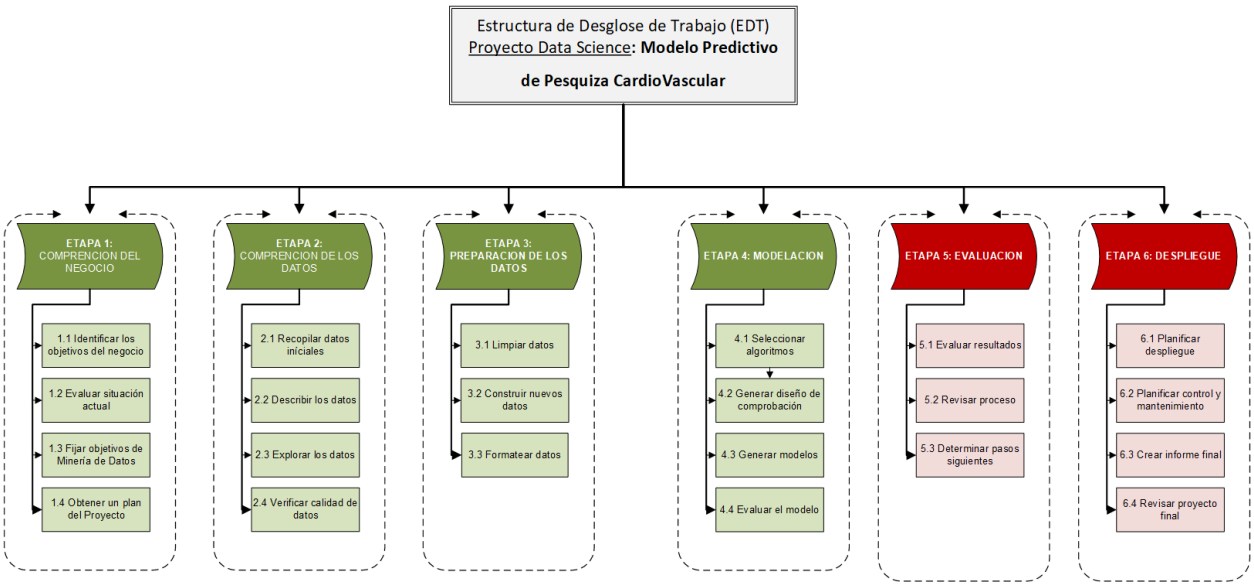


Figure 3: Estructura de Desglose del Trabajo, para proyecto de Data Science.)

2 ETAPA 2: Compresión de los datos.

2.1 Recopilación de datos iniciales:

Base de Datos de la población en control, en la Comuna de Quellón, con corte a junio 2017, con un total de 2.865 registros y 41 campos y Base de datos EMP del año 2016 y a junio 2017, extraídos del sistema RAYEN (apartada por el Subdepartamento de Tecnología de la Información del Servicio de Salud Chiloé), con un total de 2.436 registros y 68 campos. Registros totales entre ambas bases de datos fueron 5.301.

2.2 Descripción de datos.

Se llegó a la fase de modelado con una matriz depurada que contenía nueve campos predictores. Estos fueron: edad, circunferencia de cintura (CC_CM), presión arterial sistólica, colesterol, talla, presión arterial diastólica, peso, sexo, y tabaquismo. Se dispuso de 3.586 registros, 2.006 correspondientes al grupo que presenta al menos una de las tres patologías estudiadas (Grupo SI) y 1.580 a la categoría que eventualmente no presenta patología (Grupo NO).

La **Variable Objetivo** se definió con la denominación "PVC", compuesta por dos grupos: GRUPO SI = Grupo de pacientes en control del Programa Cardiovascular, que presenta al menos una de las tres patologías,

DM HTA o DLP. GRUPO NO = Grupo de Pacientes EMPA (2016 a junio 2017) y que no están en control en Programa Cardiovascular, al corte de junio del 2017 y, eventualmente, no presenta ninguna de las tres patologías señaladas. Luego, este grupo servirá para poder discriminar y encontrar aquellos patrones en los datos que caracterizan a las personas con algunas de las tres patologías del grupo "SI" y las diferencian de aquellos en el grupo "NO".

2.3 Exploración de datos.

Se analizara el problema de los Factores de Riesgo Cardiovascular Mayores desde una perspectiva de procesos y se estudiaron las técnicas que permiten descubrir el conocimiento del fenómeno almacenado en las bases de datos de la Población en Control cardiovascular que presenta DM II, HTA o DLP en exámenes de medicina preventiva del adulto (EMPA). Se identificaran patrones contenidos en los datos para determinar las variables predictivas y seleccionar los algoritmos de Machine Learning que se utilizaran en el desarrollo del modelo de pesquiza. Se desarrollara un prototipo funcional del modelo de Machine Learning, finalmente, evaluar la calidad de predicción del prototipo y corregir los posibles errores.

2.3.1 Importación de matriz de datos.

```
## Importación de dataframe
datos <- read_excel("PVC_CCM.xlsx")
```

2.3.2 Análisis de estructura de matriz datos.

```
glimpse(datos)
```

```
## Rows: 3,058
## Columns: 9
## $ PCV                          <chr> "SI", "SI", "SI", "SI", "SI", "SI", "SI", ~
## $ SEXO                         <chr> "M", "F", "F", "F", "M", "F", "M", "M", "F~
## $ EDAD_AÑOS                    <dbl> 56, 81, 60, 84, 76, 79, 70, 51, 44, 66, 62~
## $ PESO_KG                      <dbl> 110.2, 70.0, 92.4, 70.2, 77.1, 57.9, 113.0~
## $ TALLA_CM                     <dbl> 168, 144, 155, 152, 150, 149, 167, 160, 14~
## $ CC_CM                        <dbl> 119.0, 97.0, 110.0, 113.0, 107.0, 99.5, 12~
## $ PRESION_ARTERIAL_SISTOLICA   <dbl> 126, 130, 180, 116, 180, 110, 140, 140, 10~
## $ PRESION_ARTERIAL_DIASTOLICA <dbl> 80, 60, 86, 70, 80, 60, 72, 80, 70, 80, 78~
## $ COLESTEROL_TOTAL             <dbl> 275, 171, 216, 198, 223, 235, 241, 203, 15~
```

2.3.3 Visualización de matriz de datos.

```
datos <-tibble(datos)
datos
```

```
## # A tibble: 3,058 x 9
##    PCV   SEXO  EDAD_AÑOS PESO_KG TALLA_CM CC_CM PRESION_ARTERIAL_SISTOLICA
##    <chr> <chr>     <dbl>   <dbl>    <dbl> <dbl>                       <dbl>
##  1 SI    M            56   110.       168 119                           126
##  2 SI    F            81    70        144  97                           130
```

```
##  3 SI    F            60    92.4      155 110                          180
##  4 SI    F            84    70.2      152 113                          116
##  5 SI    M            76    77.1      150 107                          180
##  6 SI    F            79    57.9      149  99.5                        110
##  7 SI    M            70   113        167 129                          140
##  8 SI    M            51    89.5      160 111                          140
##  9 SI    F            44    71.5      148  97                          106
## 10 SI    F            66    99.5      155 132                          128
## # i 3,048 more rows
## # i 2 more variables: PRESION_ARTERIAL_DIASTOLICA <dbl>, COLESTEROL_TOTAL <dbl>
```

2.4 Verificación de calidad de datos.

```
# Resumen de estadisticos de variables cuantitativas y categoricas #

skim(datos)
```

Table 1: Data summary

Name	datos
Number of rows	3058
Number of columns	9
Column type frequency:	
character	2
numeric	7
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
PCV	0	1	2	2	0	2	0
SEXO	0	1	1	1	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
EDAD_AÑOS	10	1	50.67	15.14	19	39	50	60	94	
PESO_KG	0	1	76.63	13.48	40	67	75	85	125	
TALLA_CM	0	1	158.00	8.73	135	151	157	164	192	
CC_CM	3	1	100.42	11.17	65	93	100	107	140	
PRESION_ARTERIAL_SISTOLICA	6	1	121.78	16.37	80	110	120	130	190	
PRESION_ARTERIAL_DIASTOLICA	4	1	74.95	9.88	48	70	78	80	110	
COLESTEROL_TOTAL	0	1	200.19	35.35	130	174	196	222	343	

3 ETAPA 3: Preparación de los Datos.

3.1 Limpieza de datos

3.1.1 Renombrar variables.

```
# Renombrar variables, de manera mas compacta, usando dplyr

datos <- datos %>%
  # Se usa el operador pipe (%>%) para encadenar operaciones sobre el data frame 'datos'
  rename(
    EDAD   = EDAD_AÑOS,                    # Renombra la variable 'EDAD_AÑOS' como 'EDAD'
    PESO   = PESO_KG,                      # Renombra 'PESO_KG' como 'PESO'
    TALLA  = TALLA_CM,                     # Renombra 'TALLA_CM' como 'TALLA'
    CC     = CC_CM,                        # Renombra 'CC_CM' como 'CC' (Circunferencia de Cintura)
    PAS    = PRESION_ARTERIAL_SISTOLICA,   # Renombra 'PRESION_ARTERIAL_SISTOLICA' como 'PAS'
    PAD    = PRESION_ARTERIAL_DIASTOLICA,  # Renombra 'PRESION_ARTERIAL_DIASTOLICA' como 'PAD'
    COLTRL = COLESTEROL_TOTAL              # Renombra 'COLESTEROL_TOTAL' como 'COLTRL'
  )

# Verificar que los nombres de las columnas hayan sido cambiados correctamente
names(datos)  # Muestra el vector de nombres de columna del data frame 'datos'
```

```
## [1] "PCV"    "SEXO"   "EDAD"   "PESO"   "TALLA"  "CC"     "PAS"    "PAD"
## [9] "COLTRL"
```

3.1.2 transformación de variables categóricas a “Factor”.

```
## Transformar variables categoricas a factor

datos <- datos %>%
    mutate(
       across(
      where(is.character),  # Selecciona columnas que son de tipo 'character'
      as.factor             # Convierte esas columnas a tipo 'factor'
    )
  )

# Visualizar la estructura del data frame para confirmar los cambios
glimpse(datos)  # Muestra un resumen de cada columna
```

```
## Rows: 3,058
## Columns: 9
## $ PCV    <fct> SI, SI, SI, SI, SI, SI, SI, SI, SI, SI, SI, SI, SI, SI, SI, SI, SI,~
## $ SEXO   <fct> M, F, F, F, M, F, M, M, F, F, F, F, F, F, F, F, M, F, M, M, M, ~
## $ EDAD   <dbl> 56, 81, 60, 84, 76, 79, 70, 51, 44, 66, 62, 46, 58, 46, 52, 34,~
## $ PESO   <dbl> 110.2, 70.0, 92.4, 70.2, 77.1, 57.9, 113.0, 89.5, 71.5, 99.5, 7~
## $ TALLA  <dbl> 168, 144, 155, 152, 150, 149, 167, 160, 148, 155, 147, 154, 160~
## $ CC     <dbl> 119.0, 97.0, 110.0, 113.0, 107.0, 99.5, 129.0, 111.0, 97.0, 132~
## $ PAS    <dbl> 126, 130, 180, 116, 180, 110, 140, 140, 106, 128, 120, 100, 150~
```

```
## $ PAD    <dbl> 80, 60, 86, 70, 80, 60, 72, 80, 70, 80, 78, 60, 80, 62, 70, 64,~
## $ COLTRL <dbl> 275, 171, 216, 198, 223, 235, 241, 203, 156, 183, 214, 216, 216~
```

3.1.3 Identificación de asociaciones entre variables, y valores atípicos.

A continuación, cinco evidencias clave que aporta la "Matriz de Gráficos según PCV" en relación con la población bajo estudio:

- **Edad significativamente mayor en pacientes con evento cardiovascular (PCV = Sí)** Las curvas de densidad de edad (diagonal) muestran un claro desplazamiento hacia rangos superiores en el grupo PCV = Sí, con mediana alrededor de 60–65 años, frente a 40–45 años en el grupo PCV = No. Esto confirma que la edad es un importante factor de riesgo asociado a la aparición de eventos cardiovasculares.
- **Mayor masa corporal en el grupo PCV = Sí** En el histograma y densidad de peso, el grupo con PCV presenta una distribución desplazada hacia valores más altos (mediana 80 kg vs 70 kg), indicando una mayor prevalencia de exceso de peso u obesidad, condicionante conocido de riesgo cardiovascular.
- **Incremento de la circunferencia de cintura (CC) en pacientes con PCV** La densidad de CC evidencia valores medios superiores en PCV = Sí (aprox. 100 cm) comparado con PCV = No (90 cm). Esta medida de adiposidad central refuerza su vínculo con el riesgo cardiometabólico y la necesidad de intervenciones dirigidas a la reducción de grasa abdominal.
- **Presión arterial sistólica (PAS) más elevada en el grupo PCV = Sí** El box-plot y la densidad de PAS revelan una mediana cercana a 140 mmHg en PCV = Sí frente a 125 mmHg en PCV = No. Este hallazgo subraya la hipertensión sistólica como factor pronóstico primario que debería monitorizarse y controlarse en la Atención Primaria.
- **Concentración de colesterol total más alta en pacientes con PCV** La distribución de colesterol total se desplaza hacia la derecha en el grupo PCV = Sí, con un maximo de densidad en torno a 240 mg/dL, mientras que en PCV = No se sitúa cerca de 200 mg/dL. Esto evidencia la hipercolesterolemia como determinante clave en la fisiopatología de la enfermedad cardiovascular y un objetivo prioritario de los programas de prevención.

```r
# Crear el gráfico con color por categoría PCV

p <- ggpairs(
  datos,
  columns = 1:9,


  mapping = aes(color = PCV),

  # Configura la parte inferior de la matriz: dispersogramas
  lower = list(continuous = wrap("points", alpha = 0.6, size = 1)),

  # Parte superior de la matriz: coeficientes de correlación
  upper = list(continuous = wrap("cor", size = 2.5)),

  # Diagonal: densidades para variables continuas
  diag = list(continuous = wrap("densityDiag", alpha = 0.5)),

  # Título del gráfico
```

```
  title = "Matriz de Gráficos, según PCV"
)

# Ajustar tamaño de letra en ejes y título
p <- p + theme(
  axis.text.x = element_text(size = 5),      # Texto de eje X
  axis.text.y = element_text(size = 7),      # Texto de eje Y
  strip.text = element_text(size = 7),       # Texto de los encabezados de las facetas
  plot.title = element_text(size = 12, hjust = 0.5)  # Título centrado y más grande
)

# Mostrar el gráfico
p
```

3.1.4 Tratamiento de datos perdidos (NA) y Atípicos (Outliers).

El *Predictive mean matching* o *emparejamiento predictivo de medias*, calcula el valor previsto de la variable objetivo Y Según el modelo de imputación especificado. Para cada entrada faltante, el método forma un pequeño conjunto de donantes candidatos (normalmente de 3, 5 o 10 miembros) a partir de todos los casos completos cuyos valores predichos se aproximan al valor predicho para la entrada faltante. Se extrae aleatoriamente un donante entre los candidatos y se utiliza su valor observado para reemplazar el valor faltante. Se asume que la distribución de la celda faltante coincide con los datos observados de los donantes candidatos.

El emparejamiento predictivo de medias es un método fácil de usar y versátil. Es bastante robusto a las transformaciones de la variable objetivo, por lo que la imputación registro (Y) A menudo produce resultados similares a la imputación exp(Y). El método también permite variables objetivo discretas. Las imputaciones se basan en valores observados en otros lugares, por lo que son realistas. No se producirán imputaciones fuera del rango de datos observados, lo que evita problemas con imputaciones sin sentido (p. ej., altura negativa). El modelo es implícito (Little y Rubin, 2002) , lo que significa que no es necesario definir un modelo explícito para la distribución de los valores faltantes (Fuente: https://stefvanbuuren.name/fimd/sec-pmm.html?)

3.1.5 Desarrollo de Función para automatizar la Imputación de Nulos e Identificación y Tratamiento de datos Atípicos (*Outliers*).

A continuación se describe la secuencia de tareas realizadas por la función desarrollada, denominada **imputar_completo_con_outliers**:

- Se identifican y separan las variables numéricas y categóricas del conjunto de datos.
- Se elabora un resumen de valores faltantes (NA) en las variables cuantitativas para evaluar su magnitud.
- Se aplica una imputación inicial de las variables numéricas mediante el método de predicción por correspondencia empírica (pmm) con mice.
- Sobre los datos imputados, se detectan outliers según el criterio de rango intercuartílico (IQR) y se reemplazan por NA.
- Se genera un informe de la cantidad y porcentaje de outliers convertidos en NA por variable.
- Se reconstruye el dataset uniendo las variables numéricas (ahora con NA en outliers) y las variables categóricas originales.

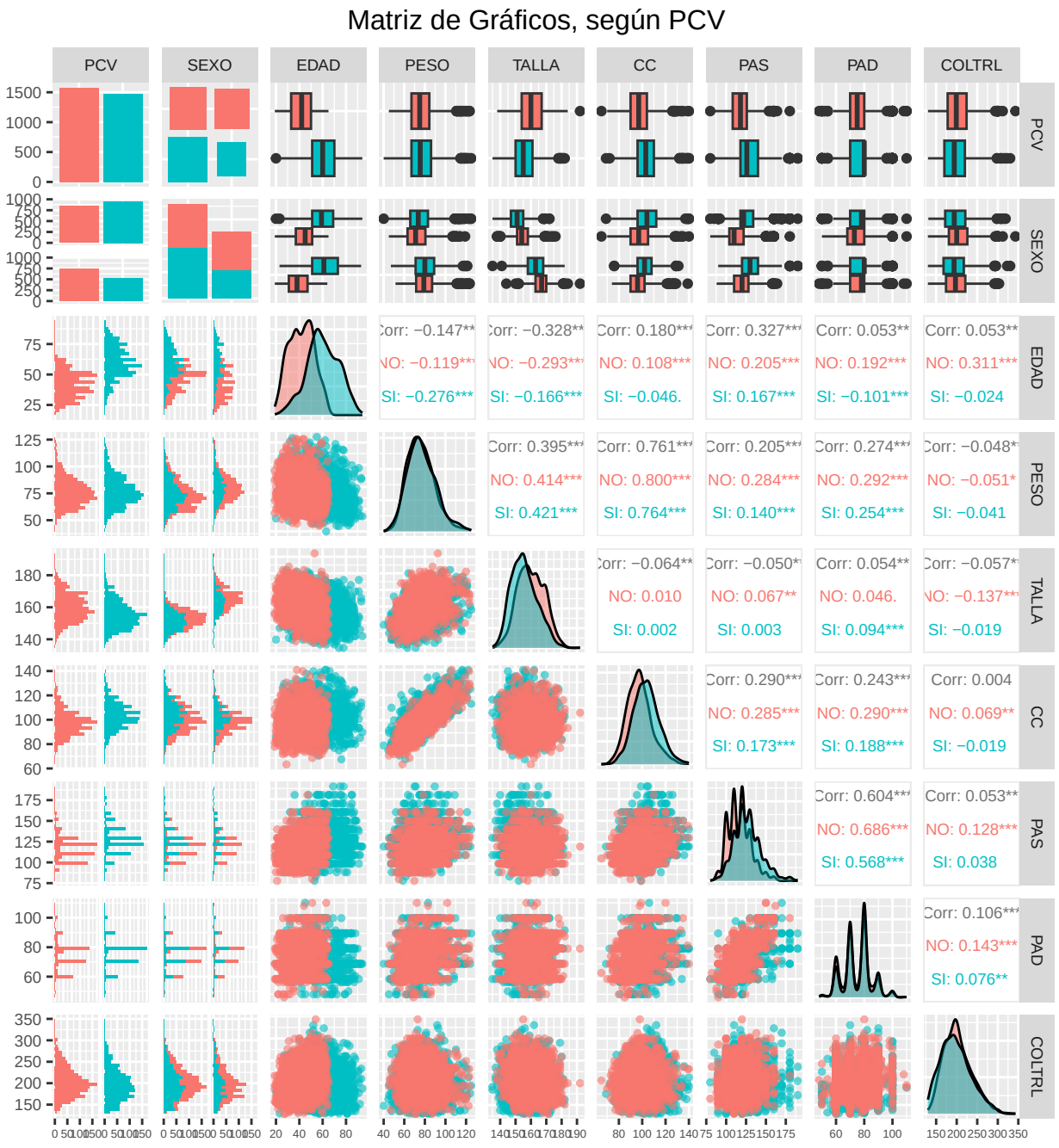


Figure 4: Resumen de herramientas gráficas e indicadores de correlación de variables en analisis.

- Se definen métodos de imputación por variable: pmm para numéricas y polinómica/logística para categóricas.
- Se realiza una imputación final conjunta de todas las variables con mice, obteniendo un dataset completo listo para análisis y modelado.

Seguidamente se presenta el código de la misma:

```
imputar_completo_con_outliers <- function(datos) {
  library(dplyr)
  library(tidyr)
  library(mice)
  library(purrr)
  library(tibble)

  # 1. Separar variables numéricas y categóricas
  vars_numericas <- names(datos)[sapply(datos, is.numeric)]
  vars_categoricas <- names(datos)[!sapply(datos, is.numeric)]

  # 2. Tabla resumen de NA's en variables cuantitativas
  resumen_na <- datos %>%
    summarise(across(all_of(vars_numericas), ~ sum(is.na(.)))) %>%
    pivot_longer(cols = everything(), names_to = "variable",
                 values_to = "n_NA") %>%
    mutate(porc_NA = round(n_NA / nrow(datos) * 100, 2))

  print("Resumen de NA's en variables cuantitativas:")
  print(resumen_na)

  # 3. Imputación inicial (solo para cuantitativas)
  set.seed(123)
  imp1 <- mice(datos[vars_numericas],
               method = "pmm", m = 1, maxit = 5, print = FALSE)
  datos_cuant_imputados <- complete(imp1)

  # 4. Reemplazar outliers en cuantitativas por NA
  datos_outliers_na <- datos_cuant_imputados %>%
    mutate(across(
      everything(),
      ~ {
        q1 <- quantile(., 0.25, na.rm = TRUE)
        q3 <- quantile(., 0.75, na.rm = TRUE)
        iqr <- q3 - q1
        .[. < (q1 - 1.5 * iqr) | . > (q3 + 1.5 * iqr)] <- NA
        .
      }
    ))

  # 5. Tabla resumen de outliers por variable
  resumen_outliers <- map_dfr(vars_numericas, function(var) {
    original <- datos_cuant_imputados[[var]]
    con_na <- datos_outliers_na[[var]]
    tibble(variable = var,
           n_outliers = sum(is.na(con_na) & !is.na(original)),
```

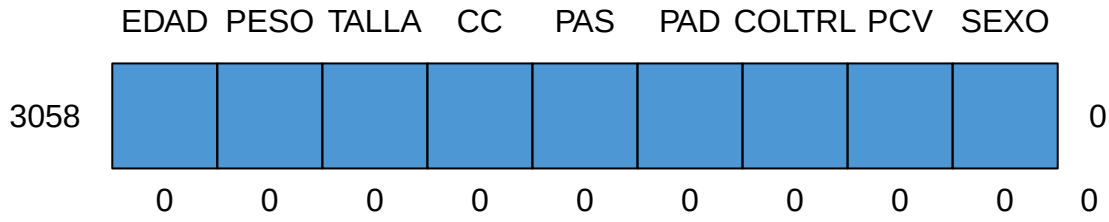
```
          porc_outliers = round(sum(is.na(con_na) & !is.na(original))
                                / nrow(datos) * 100, 2))
  })

  print("Resumen de outliers convertidos a NA:")
  print(resumen_outliers)

  # 6. Reconstruir dataset con categóricas + cuantitativas con outliers NA
  datos_completo <- bind_cols(
    datos_outliers_na,
    datos[vars_categoricas]
  )

  # 7. Imputación final (categoricas y cuantitativas)
  metodos <- make.method(datos_completo)
  metodos[vars_numericas] <- "pmm"
  metodos[vars_categoricas] <- sapply(
    datos_completo[vars_categoricas], function(x) {
    if (n_distinct(x) == 2) "logreg" else "polyreg"
  })

  set.seed(456)
  imp2 <- mice(datos_completo, method = metodos, m = 1, maxit = 5, print = FALSE)
  datos_final <- complete(imp2) %>% as_tibble()

  # Diagnóstico final opcional
  print("Patrones de NA luego de imputación final:")
  print(md.pattern(datos_final))

  return(datos_final)
}
```

3.1.6 Aplicación de la Función para automatizar la Imputacion de Nulos y Identificación y Tratamiento de datos Atípicos (*Outliers*).

```
# Aplicación de la funcion sobre datos.
datos_final <- imputar_completo_con_outliers(datos)
```

```
## [1] "Resumen de NA's en variables cuantitativas:"
## # A tibble: 7 x 3
##   variable  n_NA porc_NA
##   <chr>    <int>   <dbl>
## 1 EDAD        10    0.33
## 2 PESO         0    0
## 3 TALLA        0    0
## 4 CC           3    0.1
## 5 PAS          6    0.2
## 6 PAD          4    0.13
## 7 COLTRL       0    0
## [1] "Resumen de outliers convertidos a NA:"
## # A tibble: 7 x 3
```

```
##   variable n_outliers porc_outliers
##   <chr>         <int>         <dbl>
## 1 EDAD              4          0.13
## 2 PESO             47          1.54
## 3 TALLA             2          0.07
## 4 CC               44          1.44
## 5 PAS              41          1.34
## 6 PAD             111          3.63
## 7 COLTRL           26          0.85
## [1] "Patrones de NA luego de imputación final:"
##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```




```
##      EDAD PESO TALLA CC PAS PAD COLTRL PCV SEXO
## 3058    1    1     1  1   1   1      1   1    1 0
##         0    0     0  0   0   0      0   0    0 0
```

3.1.7 Visualización de variables cuantitativas en función de la variable objetivo (PCV), con tratamiento e imputación de nulos y atípicos (*Outliers*).

```
variables <- c("EDAD", "PESO", "TALLA", "CC", "PAS", "PAD", "COLTRL")
plot_list <- list()

# Crear un gráfico por variable
for (var in variables) {
  p <- ggplot(datos_final, aes(x = PCV, y = .data[[var]], fill = PCV)) +
    geom_boxplot(alpha = 0.7, position = position_dodge(width = 0.75)) +
    labs(
      title = paste("Distribución de", var),
      x = "PCV",
      y = var
    ) +
    scale_fill_brewer(palette = "Set2") +  # Puedes elegir otra paleta si prefieres
    theme_minimal(base_size = 12) +
    theme(legend.position = "none")  # Oculta leyenda en cada gráfico individual

  plot_list[[var]] <- p
}

# Combinar en una matriz 4x2 y agregar título general
combined_plot <- wrap_plots(plot_list, ncol = 2) +
  plot_annotation(title = "Distribución de variables cuantitativas con tratamiento de imputación de dat
  theme = theme(plot.title = element_text(size = 12, face = "bold", hjust = 0.5)))

# Mostrar
combined_plot
```

Tras el tratamiento de valores faltantes y atípicos, la distribución de los principales factores de riesgo mantiene patrones claros entre quienes presentan evento cardiovascular (PCV = Sí) y quienes no (PCV = No). Entre las evidencias más relevantes destacan:

Edad más elevada en PCV = Sí – Mediana de edad 60 años frente a 42 años en el grupo sin evento. – IQR (55–70) vs (35–50), lo que confirma a la edad como factor de riesgo primordial.

Mayor índice de adiposidad central (CC) – Mediana de circunferencia de cintura 105 cm en PCV = Sí vs 100 cm en PCV = No. – Refuerza la asociación entre adiposidad abdominal y riesgo cardiometabólico.

Presión arterial sistólica (PAS) incrementada – Mediana de PAS 125 mmHg en PCV = Sí frente a 120 mmHg en quienes no tuvieron evento. – Muestra la persistencia de la hipertensión sistólica como predictor de eventos.

Peso ligeramente superior en el grupo con evento – Mediana de peso 78 kg en PCV = Sí vs 75 kg en PCV = No. – A pesar de la imputación y depuración, el exceso de peso se mantiene como cofactor.

Colesterol total menor en PCV = Sí – Mediana de colesterol 190 mg/dL en PCV = Sí vs 200 mg/dL en PCV = No. – Indica posible efecto de intervenciones farmacológicas (estatinas) tras la ocurrencia del evento.

Estos hallazgos subrayan la necesidad de focalizar estrategias de prevención primaria en la población de mayor edad y con marcadores de riesgo (adiposidad central, hipertensión), así como reforzar el seguimiento y adherencia a tratamientos hipolipemiantes en quienes ya han sufrido un evento cardiovascular.

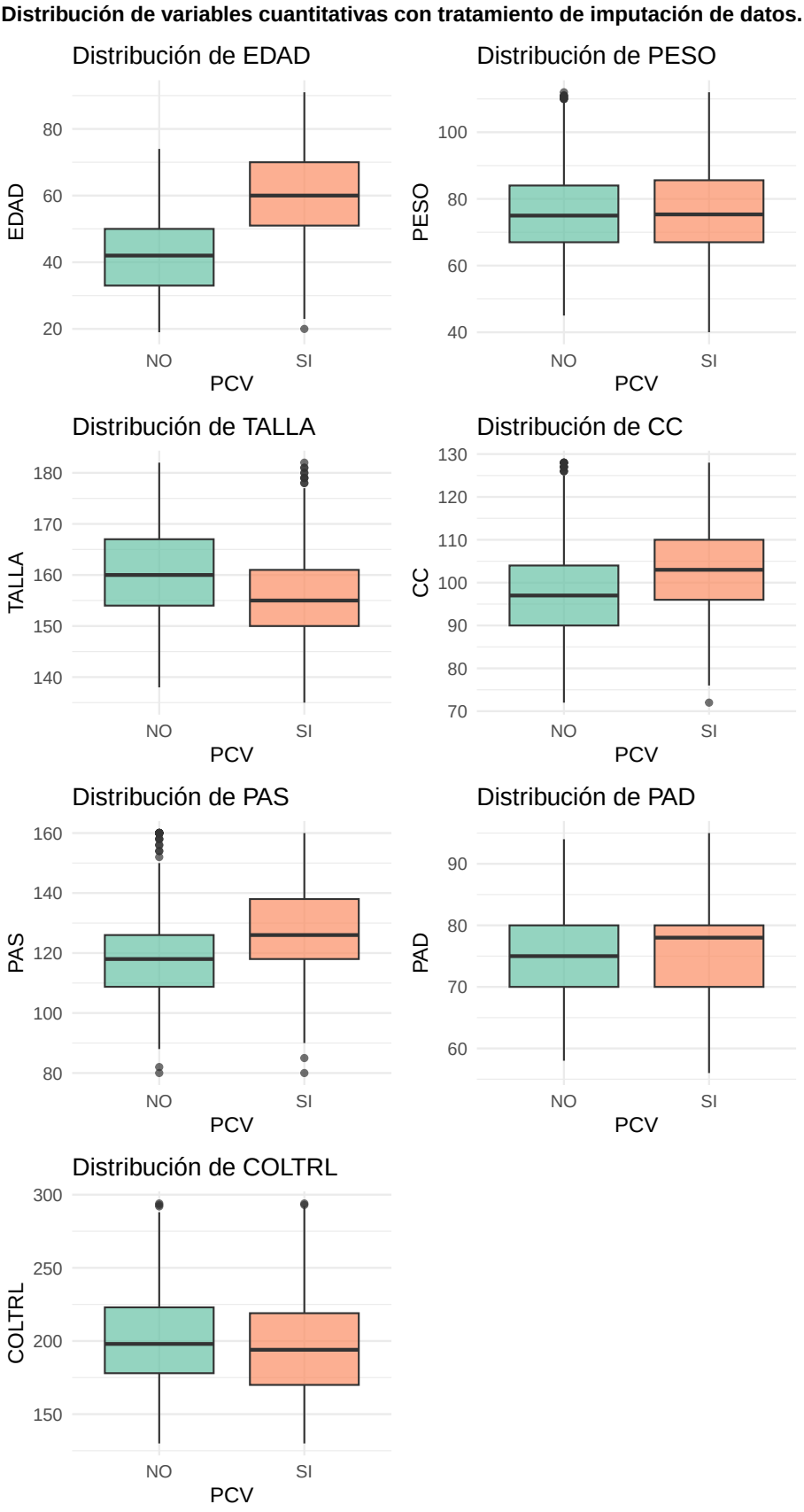


Figure 5: Imputación de datos perdidos y datos atípicos (Outliers)

3.2 Construcción de nuevos datos.

La incorporación conjunta de IMC, ICT y PAM mejora la sensibilidad y especificidad del modelo de predicción de riesgo cardiovascular al capturar distintos ejes de la fisiopatología:

- Adiposidad general (IMC)
- Obesidad central y carga metabólica (ICT)
- Carga hemodinámica crónica (PAM)

Estos predictores son de bajo costo, fácilmente medibles en todos los niveles de atención (APS, Urgencias, Hospital) y permiten optimizar la gestión de listas de espera, la priorización en pabellones quirúrgicos para intervenciones relacionadas (angioplastias, by-pass) y la asignación de recursos diagnósticos y terapéuticos en la Red Asistencial. De esta manera, se contribuye a una atención más oportuna y efectiva de la población en control por enfermedades cardiovasculares crónicas.

```
datos_final <- datos_final %>%
  mutate(IMC=PESO/((TALLA/100)^2),   ##Indice de Masa Corporal
         ICT= CC/TALLA,              ##Indice Cintura Talla
         PAM= PAS + (2*PAD)/3)       ##Presion Arterial Media
head(datos_final)
```

```
## # A tibble: 6 x 12
##    EDAD  PESO TALLA    CC   PAS   PAD COLTRL PCV   SEXO    IMC   ICT   PAM
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl> <fct> <fct> <dbl> <dbl> <dbl>
## 1    56 110.    168 119     126    80    275 SI    M      39.0 0.708  179.
## 2    81  70     144  97     130    60    171 SI    F      33.8 0.674  170
## 3    60  92.4   155 110     150    86    216 SI    F      38.5 0.710  207.
## 4    84  70.2   152 113     116    70    198 SI    F      30.4 0.743  163.
## 5    76  77.1   150 107     160    80    223 SI    M      34.3 0.713  213.
## 6    79  57.9   149  99.5   110    60    235 SI    F      26.1 0.668  150
```

3.3 Formateo de datos.

3.3.1 Partición de datos.

```
set.seed(123)
# Partición estratificada 80% entrenamiento / 20% test
indices <- createDataPartition(datos_final$PCV, p = 0.8, list = FALSE)
train_df <- datos_final[indices, ]
test_df  <- datos_final[-indices, ]
```

3.3.2 Escalado y centrado de datos de variables cuantitativas.

```
# Ejemplo de escalado y centrado previo
pp <- preProcess(train_df[, -which(names(train_df)=="PCV")],
                 method = c("center", "scale"))
train_pp <- predict(pp, train_df)
test_pp  <- predict(pp, test_df)
```

3.3.3 Creación de variables *Dummy* para atributos categóricos.

```
# Identificar variables categóricas en el conjunto de entrenamiento
cat_vars <- names(train_pp)[ sapply(train_pp, function(x) is.factor(x) || is.character(x)) ]
cat_vars
```

```
## [1] "PCV"  "SEXO"
```

```
dv_all <- dummyVars(
  formula  = ~ .,
  data     = train_pp,
  fullRank = TRUE,
  sep      = "_",
)

# Aplicar la transformación a train_pp y test_pp
train_dummy <- predict(dv_all, newdata = train_pp)
test_dummy  <- predict(dv_all, newdata = test_pp)

# Convertir matrices a data.frames
train_dummy <- as.data.frame(train_dummy)
test_dummy  <- as.data.frame(test_dummy)
```

4 ETAPA 4: Modelado

##Selección de Algoritmos Para seleccionar el modelo más adecuado entre **regresión logística binaria** y **XGBoost** en un problema de clasificación de riesgo cardiovascular, es esencial comparar aspectos de interpretabilidad, rendimiento predictivo, robustez de los datos, requerimientos computacionales y aplicabilidad en el contexto de salud pública. La regresión logística ofrece una implementación sencilla y coeficientes directamente interpretables como odds ratio, lo cual facilita la adopción de resultados por parte de clínicos y gestores sanitarios. Por su parte, XGBoost incorpora métodos de ensamblado de árboles con regularización, que suelen superar en exactitud a los modelos lineales y manejan automáticamente valores faltantes y atípicos, aunque a costa de mayor complejidad computacional.

4.1 Generación de Diseño de Comprobación.

Un diseño de comprobación basado en un split estratificado permite estimar la generalización de los modelos en datos no vistos, evitando sesgos de sobreajuste. Al reservar un conjunto de prueba independiente y usar una semilla fija se garantiza la reproducibilidad y la imparcialidad en la comparación entre regresión logística y XGBoost. La proporción recomendada de 80 % para entrenamiento y 20 % para prueba equilibra la variabilidad de la estimación y la cantidad de datos disponibles para ajuste de hiperparámetros mediante validación cruzada interna. Este enfoque aísla claramente la fase de evaluación final, mejorando la transparencia y confiabilidad de la selección de modelo en contextos de salud pública.

4.2 Generación y Evaluación de Modelos de Machine Learning.

4.2.1 Regresión logistica lineal.

```r
##Formulación de modelo inicial utilizando todas las variables

log_1<-glm(PCV_SI~.,data = train_dummy, family="binomial")
summary(log_1)
```

```
##
## Call:
## glm(formula = PCV_SI ~ ., family = "binomial", data = train_dummy)
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.01990    0.08518  -0.234  0.81530
## EDAD         1.80769    0.08945  20.209  < 2e-16 ***
## PESO         2.75904    0.91494   3.016  0.00257 **
## TALLA        0.03001    0.51008   0.059  0.95309
## CC          -3.53129    1.47216  -2.399  0.01645 *
## PAS          0.48330    0.07447   6.490 8.59e-11 ***
## PAD         -0.35184    0.07195  -4.890 1.01e-06 ***
## COLTRL      -0.31215    0.05478  -5.698 1.21e-08 ***
## SEXO_M       0.04702    0.16282   0.289  0.77273
## IMC         -2.43794    0.86848  -2.807  0.00500 **
## ICT          4.53431    1.69658   2.673  0.00753 **
## PAM               NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3389.6  on 2446  degrees of freedom
## Residual deviance: 2086.6  on 2436  degrees of freedom
## AIC: 2108.6
##
## Number of Fisher Scoring iterations: 5
```

Nuestro modelo ha obtenido un AIC de 2108.6. El AIC cuantifica la cantidad relativa de información que se pierde al ajustar un modelo: cuanto menor es esa pérdida, mayor la calidad del ajuste. Muchas de las variables que hemos incluido no resultan significativas (p-value $>$ 0,05) y, por tanto, deben descartarse. Para optimizar la selección de variables y reducir el AIC, podemos emplear la función stepAIC() de la librería MASS, que elimina automáticamente las variables no informativas y retorna el modelo con el menor AIC posible.

```r
#Selección del modelo
stepAIC(log_1,trace=T)
```

```
## Start:  AIC=2108.62
## PCV_SI ~ EDAD + PESO + TALLA + CC + PAS + PAD + COLTRL + SEXO_M +
##     IMC + ICT + PAM
##
##
## Step:  AIC=2108.62
## PCV_SI ~ EDAD + PESO + TALLA + CC + PAS + PAD + COLTRL + SEXO_M +
##     IMC + ICT
```

```
##
##          Df Deviance    AIC
## - TALLA   1   2086.6 2106.6
## - SEXO_M  1   2086.7 2106.7
## <none>        2086.6 2108.6
## - CC      1   2092.4 2112.4
## - ICT     1   2093.8 2113.8
## - IMC     1   2094.5 2114.5
## - PESO    1   2095.7 2115.7
## - PAD     1   2111.2 2131.2
## - COLTRL  1   2120.0 2140.0
## - PAS     1   2130.3 2150.3
## - EDAD    1   2728.5 2748.5
##
## Step:  AIC=2106.62
## PCV_SI ~ EDAD + PESO + CC + PAS + PAD + COLTRL + SEXO_M + IMC +
##     ICT
##
##          Df Deviance    AIC
## - SEXO_M  1   2086.7 2104.7
## <none>        2086.6 2106.6
## - IMC     1   2094.5 2112.5
## - CC      1   2095.4 2113.4
## - PESO    1   2095.7 2113.7
## - ICT     1   2097.7 2115.7
## - PAD     1   2111.2 2129.2
## - COLTRL  1   2120.0 2138.0
## - PAS     1   2130.3 2148.3
## - EDAD    1   2730.7 2748.7
##
## Step:  AIC=2104.72
## PCV_SI ~ EDAD + PESO + CC + PAS + PAD + COLTRL + IMC + ICT
##
##          Df Deviance    AIC
## <none>        2086.7 2104.7
## - IMC     1   2094.9 2110.9
## - CC      1   2095.6 2111.6
## - PESO    1   2096.1 2112.1
## - ICT     1   2097.8 2113.8
## - PAD     1   2111.4 2127.4
## - COLTRL  1   2120.1 2136.1
## - PAS     1   2130.9 2146.9
## - EDAD    1   2745.6 2761.6

##
## Call:  glm(formula = PCV_SI ~ EDAD + PESO + CC + PAS + PAD + COLTRL +
##     IMC + ICT, family = "binomial", data = train_dummy)
##
## Coefficients:
## (Intercept)         EDAD         PESO           CC          PAS          PAD
##   -0.002288     1.810402     2.788631    -3.482022     0.484781    -0.352272
##      COLTRL          IMC          ICT
##   -0.312428    -2.464059     4.473741
##
```

```
## Degrees of Freedom: 2446 Total (i.e. Null);  2438 Residual
## Null Deviance:       3390
## Residual Deviance: 2087  AIC: 2105
```

El modelo de **regresión logística** identificó la edad, la presión arterial sistólica y el índice cintura–talla como predictores significativamente asociados al riesgo de evento cardiovascular, mientras que la dirección inversa de circunferencia de cintura, presión diastólica, colesterol total e IMC sugiere colinealidad y efectos de intervenciones terapéuticas posteriores. La elevada magnitud del ICT refuerza su valor como marcador de adiposidad central en evaluaciones clínicas. Estos resultados respaldan la implementación de estrategias preventivas en la Red APS centradas en el control de hipertensión y adiposidad en población de mayor edad. Se recomienda profundizar en el análisis de multicolinealidad y validar externamente el modelo para facilitar su adopción en la práctica asistencial.

```r
log_final <- glm(
  formula = PCV_SI ~ EDAD + PESO + CC + PAS + PAD + COLTRL +
    IMC + ICT, family = "binomial", data = train_dummy)
summary(log_final)
```

```
##
## Call:
## glm(formula = PCV_SI ~ EDAD + PESO + CC + PAS + PAD + COLTRL +
##     IMC + ICT, family = "binomial", data = train_dummy)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.002288   0.055263  -0.041 0.966979
## EDAD         1.810402   0.088917  20.361  < 2e-16 ***
## PESO         2.788631   0.908773   3.069 0.002151 **
## CC          -3.482022   1.173126  -2.968 0.002996 **
## PAS          0.484781   0.074309   6.524 6.85e-11 ***
## PAD         -0.352272   0.071903  -4.899 9.62e-07 ***
## COLTRL      -0.312428   0.054768  -5.705 1.17e-08 ***
## IMC         -2.464059   0.863325  -2.854 0.004315 **
## ICT          4.473741   1.346373   3.323 0.000891 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3389.6  on 2446  degrees of freedom
## Residual deviance: 2086.7  on 2438  degrees of freedom
## AIC: 2104.7
##
## Number of Fisher Scoring iterations: 5
```

```r
# Extraer OR y sus intervalos de confianza al 95%
or_broom <- tidy(
  log_final,
  exponentiate = TRUE,    # transforma coeficientes a OR (exp(beta))
  conf.int     = TRUE     # calcula IC al 95% para los coeficientes
)

# Ver resultados
print(or_broom)
```

```
## # A tibble: 9 x 7
##   term        estimate std.error statistic  p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)   0.998     0.0553   -0.0414 9.67e- 1  0.895       1.11
## 2 EDAD          6.11      0.0889   20.4    3.74e-92  5.15        7.31
## 3 PESO         16.3       0.909     3.07   2.15e- 3  2.74       96.7
## 4 CC            0.0307    1.17     -2.97   3.00e- 3  0.00307     0.306
## 5 PAS           1.62      0.0743    6.52   6.85e-11  1.41        1.88
## 6 PAD           0.703     0.0719   -4.90   9.62e- 7  0.610       0.809
## 7 COLTRL        0.732     0.0548   -5.70   1.17e- 8  0.657       0.814
## 8 IMC           0.0851    0.863    -2.85   4.32e- 3  0.0156      0.462
## 9 ICT          87.7       1.35      3.32   8.91e- 4  6.30     1238.
```

Con el objetivo de evaluar la capacidad de generalización del modelo frente a datos no vistos, empleamos el conjunto de validación test_t y fijamos un umbral de decisión en 0,50. De este modo, podemos analizar su desempeño diagnóstico al clasificar nuevos casos según su probabilidad estimada.

```
# Realizar predicciones de probabilidad
predict_log_final <- predict(log_final,
                             newdata = test_dummy,
                             type = "response")

# Convertir las probabilidades en clases (0 o 1) usando umbral 0.5
prediccion_05 <- ifelse(predict_log_final > 0.5, 1, 0)

# Ver primeras predicciones
head(prediccion_05)
```

```
## 1 2 3 4 5 6
## 1 0 0 0 1 1
```

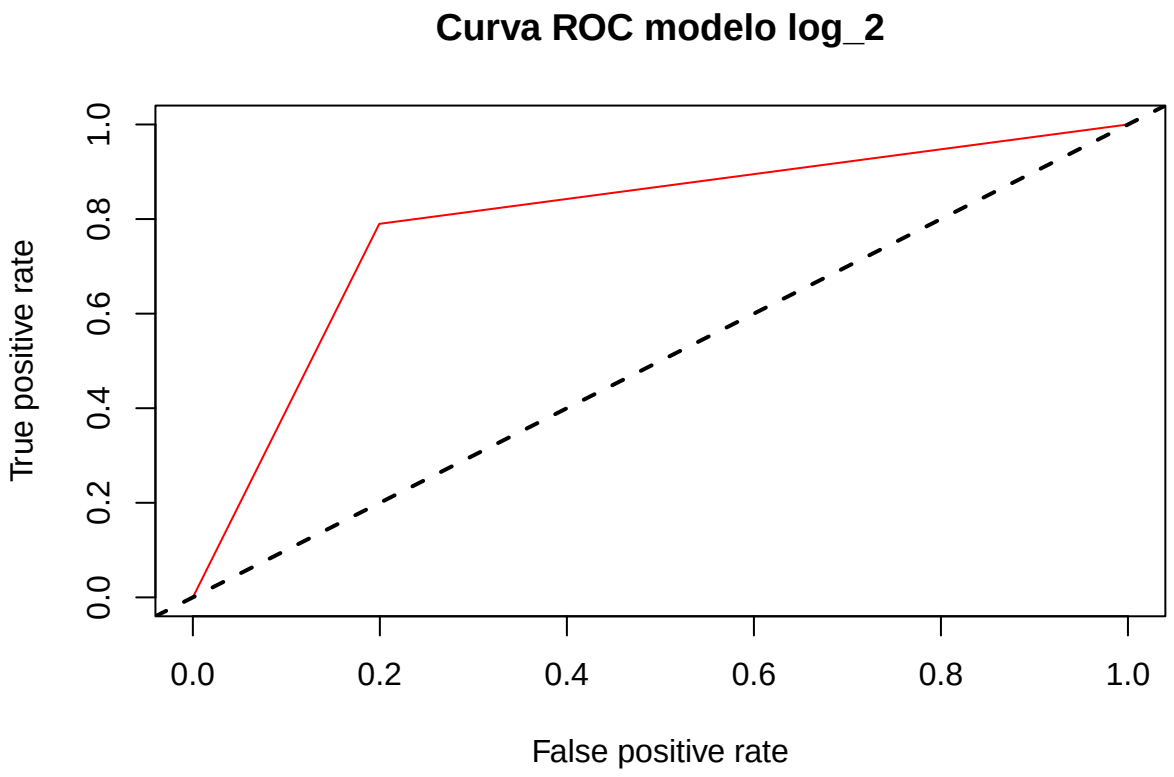
```
confusionMatrix(
  reference = factor(test_dummy$PCV_SI, levels = c(0, 1)),
  data = factor(prediccion_05, levels = c(0, 1)),
  positive = "1"
)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 253  62
##          1  63 233
##
##                Accuracy : 0.7954
##                  95% CI : (0.7612, 0.8267)
##     No Information Rate : 0.5172
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.5904
##
##  Mcnemar's Test P-Value : 1
##
```

```
##             Sensitivity : 0.7898
##             Specificity : 0.8006
##          Pos Pred Value : 0.7872
##          Neg Pred Value : 0.8032
##              Prevalence : 0.4828
##          Detection Rate : 0.3813
##    Detection Prevalence : 0.4845
##       Balanced Accuracy : 0.7952
##
##        'Positive' Class : 1
##
```

El modelo alcanza una exactitud del 79,5 % (IC 95 % 76,1–82,8) y un índice Kappa de 0,59, lo que indica concordancia moderada. La sensibilidad (78,9 %) y la especificidad (80,1 %) están equilibradas, con una precisión predictiva positiva del 78,7 % y negativa del 80,3 %, reflejando buena capacidad para identificar tanto casos como controles. La tasa de detección (38,1 %) y la prevalencia de predicción (48,5 %) confirman un ajuste adecuado al balance de clases (prevalencia real 48,3 %). En conjunto, el desempeño es sólido para su aplicación en estrategias de detección de riesgo cardiovascular en Atención Primaria.

```r
#Curva ROC
pr_05 <- prediction(prediccion_05, test_dummy$PCV_SI)
perf_log_05 <- performance(pr_05, measure = "tpr", x.measure = "fpr")
plot(perf_log_05, col = "Red", main = "Curva ROC modelo log_2")
#Diagonal o línea discriminante
abline(a=0,b=1,lwd=2,lty=2,col="black")
```



```
#AUC con umbral 0.50
 auc(test_dummy$PCV_SI,prediccion_05)
```

```
## Area under the curve: 0.7952
```

4.2.2 Métodos de Boosting: XGBoost.

Se extraen las variables dependientes u objetivo de los conjuntos de train y test.

```
#Variables dependientes
 y_train_dummy <- train_dummy$PCV_SI
 y_test_dummy <- test_dummy$PCV_SI
```

```
#Crea matriz de datos solo con variables predictoras

train_dummy_sin_PCV_SI <- train_dummy %>%
  mutate(PCV_SI = NULL)

test_dummy_sin_PCV_SI <- test_dummy %>%
  mutate(PCV_SI = NULL)
```

```
 ## comprobar que ninguna de las variables tenga varianza cero o  próxima a cero.
 nearZeroVar(train_dummy_sin_PCV_SI, saveMetrics = T)
```

```
##        freqRatio percentUnique zeroVar   nzv
## EDAD    1.223684    2.98324479   FALSE FALSE
## PESO    1.148148   17.69513690   FALSE FALSE
## TALLA   1.162162    1.92071925   FALSE FALSE
## CC      1.027778    3.75970576   FALSE FALSE
## PAS     1.155814    2.24765018   FALSE FALSE
## PAD     1.184127    1.47118921   FALSE FALSE
## COLTRL  1.088889    7.02901512   FALSE FALSE
## SEXO_M  1.437251    0.08173273   FALSE FALSE
## IMC     1.000000   73.76379240   FALSE FALSE
## ICT     1.733333   46.62852472   FALSE FALSE
## PAM     1.132353    6.45688598   FALSE FALSE
```

Una vez que todas las variables son numéricas, transformamos el conjunto de datos al formato que acepta el modelo, utilizando la función xgb.DMatrix(), donde indicamos el conjunto de datos transformado en matriz y la variable dependiente.

```
dtrain_dummy_sin_PCV_SI <- xgb.DMatrix(as.matrix(train_dummy_sin_PCV_SI),
                                       label = y_train_dummy)
dtest_dummy_sin_PCV_SI <- xgb.DMatrix(as.matrix(test_dummy_sin_PCV_SI),
                                      label = y_test_dummy)
```

Lo primero que vamos a hacer es buscar la profundidad óptima con la función expand.grid() de caret, manteniendo el resto de parámetros con los valores por defecto.

```r
grid <- expand.grid(max_depth = 1:6,
                    eta = 0.3,
                    colsample_bytree = 1,
                    gamma = 0,
                    subsample = 1,
                    min_child_weight = 1,
                    nrounds = 100)



control <- trainControl(method = "cv",
                        number = 3)


grid_model_depth <- train(x = train_dummy_sin_PCV_SI,
                        y = factor(y_train_dummy),
                        trControl = control,
                        tuneGrid = grid,
                        method = "xgbTree")

grid_model_depth
```

```
## eXtreme Gradient Boosting
##
## 2447 samples
##   11 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 1631, 1632, 1631
## Resampling results across tuning parameters:
##
##   max_depth  Accuracy   Kappa
##   1          0.7874995  0.5734237
##   2          0.7789176  0.5565990
##   3          0.7821875  0.5634424
##   4          0.7760551  0.5511589
##   5          0.7785030  0.5558873
##   6          0.7752376  0.5496764
##
## Tuning parameter 'nrounds' was held constant at a value of 100
## Tuning
##  parameter 'min_child_weight' was held constant at a value of 1
##
## Tuning parameter 'subsample' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were nrounds = 100, max_depth = 1, eta
##  = 0.3, gamma = 0, colsample_bytree = 1, min_child_weight = 1 and subsample = 1.
```

Una vez que hemos determinado la profundidad óptima (max_depth=1), podemos modificar nuestra búsqueda para afinar el resto de los hiperparámetros del modelo. En este caso, exploraremos distintos

valores de la tasa de aprendizaje (eta) y del porcentaje de observaciones muestreadas por cada árbol durante el entrenamiento (subsample).

```
 grid<-expand.grid(max_depth=1,
                     eta= c(0.025, 0.05,0.1,0.3,0.5),
                     colsample_bytree =1,
                     gamma=0,
                     subsample= c(0.3,0.5,0.8,1),
                     min_child_weight =1,
                     nrounds=100)

control<-trainControl(method="cv",
                        number=3)


grid_model_eta<-train(x=train_dummy_sin_PCV_SI,
                      y=factor(y_train_dummy),
                      trControl=control,
                      tuneGrid= grid,
                      method="xgbTree")

grid_model_eta
```

```
## eXtreme Gradient Boosting
##
## 2447 samples
##   11 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 1631, 1632, 1631
## Resampling results across tuning parameters:
##
##   eta    subsample  Accuracy   Kappa
##   0.025  0.3        0.7768666  0.5515064
##   0.025  0.5        0.7760491  0.5496504
##   0.025  0.8        0.7764581  0.5506503
##   0.025  1.0        0.7744131  0.5467538
##   0.050  0.3        0.7854460  0.5692671
##   0.050  0.5        0.7834045  0.5649761
##   0.050  0.8        0.7850365  0.5683849
##   0.050  1.0        0.7846275  0.5675055
##   0.100  0.3        0.7956634  0.5898580
##   0.100  0.5        0.7907600  0.5797989
##   0.100  0.8        0.7899425  0.5781805
##   0.100  1.0        0.7879000  0.5741863
##   0.300  0.3        0.7858600  0.5700609
##   0.300  0.5        0.7891260  0.5767943
##   0.300  0.8        0.7903505  0.5790428
##   0.300  1.0        0.7883065  0.5748615
##   0.500  0.3        0.7944329  0.5876268
##   0.500  0.5        0.7821760  0.5629159
##   0.500  0.8        0.7862615  0.5708894
```

```
##   0.500  1.0        0.7895355  0.5773924
##
## Tuning parameter 'nrounds' was held constant at a value of 100
## Tuning
##  parameter 'colsample_bytree' was held constant at a value of 1
##
## Tuning parameter 'min_child_weight' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were nrounds = 100, max_depth = 1, eta
##  = 0.1, gamma = 0, colsample_bytree = 1, min_child_weight = 1 and subsample
##  = 0.3.
```

Los valores óptimos para los parámetros que hemos indicado son eta = 0.1 y subsample = 0.3. Creamos un modelo con la función xgb.cv(), que nos permite realizar validación cruzada en XGBoost y obtener el número óptimo de iteraciones, utilizando los valores de eta y subsample obtenidos anteriormente.

```
parametros<-list(objetive="binary:logistic",
                    eval_metric="auc",
                    max_depth=1,
                    eta=0.1,
                    subsample=0.3)


 model_ini <-xgb.cv(data=dtrain_dummy_sin_PCV_SI,
                        label=y_train_dummy,
                        nrounds=1000,
                        nfold=5,
                        print_every_n=10,
                        early_stopping_rounds=50,
                        params=parametros)
```

```
## [1]  train-auc:0.770117+0.007706 test-auc:0.764897+0.017810
## Multiple eval metrics are present. Will use test_auc for early stopping.
## Will train until test_auc hasn't improved in 50 rounds.
##
## [11] train-auc:0.852287+0.004989 test-auc:0.845204+0.021930
## [21] train-auc:0.868424+0.002913 test-auc:0.863040+0.012633
## [31] train-auc:0.873266+0.002933 test-auc:0.866180+0.010304
## [41] train-auc:0.877171+0.002654 test-auc:0.868562+0.010843
## [51] train-auc:0.879637+0.002667 test-auc:0.870647+0.010294
## [61] train-auc:0.881611+0.002665 test-auc:0.871744+0.010270
## [71] train-auc:0.882910+0.002386 test-auc:0.871757+0.009912
## [81] train-auc:0.884397+0.002335 test-auc:0.872522+0.010244
## [91] train-auc:0.885637+0.002369 test-auc:0.872791+0.009903
## [101]    train-auc:0.886652+0.002258 test-auc:0.873376+0.009919
## [111]    train-auc:0.887697+0.002230 test-auc:0.872999+0.010135
## [121]    train-auc:0.888537+0.002263 test-auc:0.872398+0.010206
## [131]    train-auc:0.889346+0.002342 test-auc:0.871940+0.010189
## [141]    train-auc:0.890083+0.002198 test-auc:0.872150+0.010636
## [151]    train-auc:0.890820+0.002354 test-auc:0.872655+0.010599
## Stopping. Best iteration:
## [101]    train-auc:0.886652+0.002258 test-auc:0.873376+0.009919
```

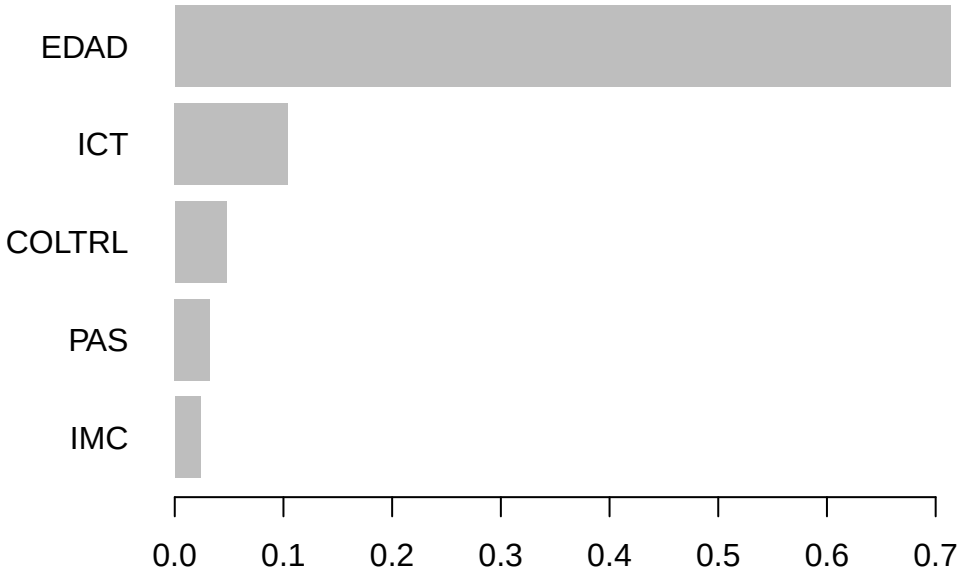
Se generó un modelo empleando la iteración óptima (101) y los parámetros determinados mediante la función expand.grid().

```
model_sub<-xgb.train(data=dtrain_dummy_sin_PCV_SI,
                     label=y_train_dummy,
                     nrounds=model_ini$best_iteration,
                     nfold=5,
                     print_every_n =10,
                     early_stopping_rounds=50,
                     params=parametros,
                     watchlist=list(val=dtest_dummy_sin_PCV_SI,
                                    train=dtrain_dummy_sin_PCV_SI))
```

```
## [23:42:40] WARNING: src/learner.cc:767:
## Parameters: { "label", "nfold", "objetive" } are not used.
##
## [1]  val-auc:0.776362    train-auc:0.775970
## Multiple eval metrics are present. Will use train_auc for early stopping.
## Will train until train_auc hasn't improved in 50 rounds.
##
## [11] val-auc:0.851947    train-auc:0.857449
## [21] val-auc:0.857868    train-auc:0.867404
## [31] val-auc:0.864932    train-auc:0.873023
## [41] val-auc:0.868580    train-auc:0.877824
## [51] val-auc:0.872527    train-auc:0.879969
## [61] val-auc:0.873836    train-auc:0.881280
## [71] val-auc:0.875869    train-auc:0.882813
## [81] val-auc:0.878154    train-auc:0.884147
## [91] val-auc:0.878030    train-auc:0.885347
## [101]    val-auc:0.879103    train-auc:0.886133
```

Obtenemos que con 101 iteraciones un AUC en train de 88.61% y en el conjunto de validación de 87,91%. Es normal conseguir una precisión menor en el conjunto de datos que en el conjunto de entrenamiento, al ser observaciones nuevas para el modelo. Buscamos las variables más importantes para el modelo que hemos realizado con la función varImp().

```
xgb_imp <- xgb.importance(feature_names = colnames(
  dtrain_dummy_sin_PCV_SI), model = model_sub)
 xgb.plot.importance(xgb_imp[1:5])
```



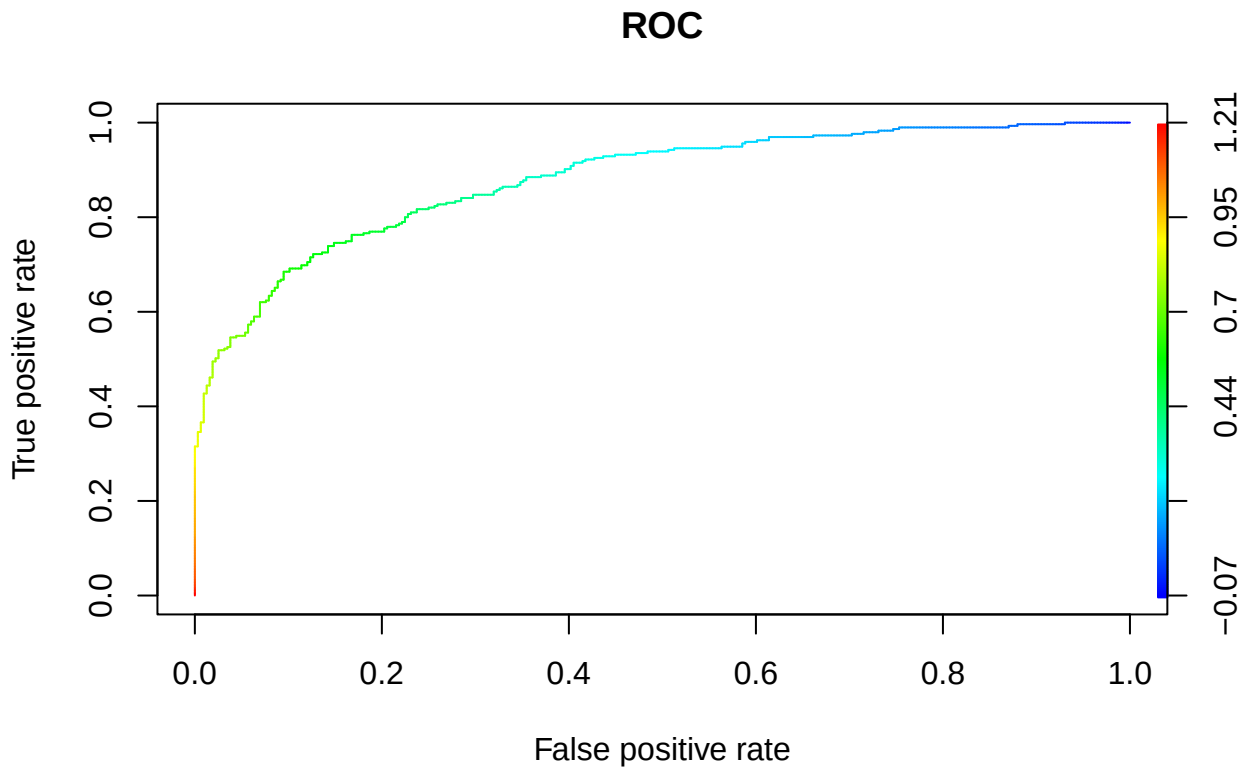

```r
#Predecimos con el modelo.
predict_xgb <- predict(model_sub, dtest_dummy_sin_PCV_SI)

predict_class <- as.factor(ifelse(predict_xgb > 0.5,1,0))

confusionMatrix(predict_class, factor(y_test_dummy), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 251  66
##          1  65 229
##
##                Accuracy : 0.7856
##                  95% CI : (0.7509, 0.8175)
##     No Information Rate : 0.5172
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.5706
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.7763
##             Specificity : 0.7943
```

```
##          Pos Pred Value : 0.7789
##          Neg Pred Value : 0.7918
##              Prevalence : 0.4828
##          Detection Rate : 0.3748
##    Detection Prevalence : 0.4812
##       Balanced Accuracy : 0.7853
##
##        'Positive' Class : 1
##
```

El modelo XGBoost alcanzó una exactitud del 78,6 % (IC 95 %: 75,1–81,8) con un de 0,57, lo que indica concordancia moderada. La sensibilidad (77,6 %) y la especificidad (79,4 %) están equilibradas, con valores predictivos positivo y negativo de 77,9 % y 79,2 %, respectivamente. La balanced accuracy de 78,5 % confirma un rendimiento uniforme independientemente del sesgo de clases (prevalencia 48,3 %). Estos indicadores sitúan a XGBoost como un instrumento sólido para la detección temprana de riesgo cardiovascular en la Red APS y el ámbito hospitalario.

```r
#CurvaROC.

pr_xgb<-prediction(as.numeric(predict_xgb),y_test_dummy)

perf_xgb<-performance(pr_xgb,measure="tpr",x.measure="fpr")

plot(perf_xgb,colorize=T,main="ROC")
```



```r
#AUCxgb
auc(y_test_dummy,as.numeric(predict_xgb))
```

```
## Area under the curve: 0.8791
```

5 Discusión y Conclusiones.

La comparación de ambos modelos revela un rendimiento global similar, con ventajas diferenciales según el contexto de aplicación. La regresión logística mostró una precisión de clasificación del 79,5 % (IC 95 %: 76,1–82,8), sensibilidad del 78,9 % y especificidad del 80,1 % (umbral 0,50), alcanzando un AUC de 0,7952 . En contraste, XGBoost obtuvo una exactitud de 78,6 % (IC 95 %: 75,1–81,8), sensibilidad del 77,6 %, especificidad del 79,4 %, un índice Kappa de 0,57 y un AUC de validación de 0,8791 . Aunque la diferencia en AUC favorece a XGBoost, ambos modelos presentan un balance adecuado entre tasa de verdaderos positivos y negativos, lo que garantiza su aplicabilidad en la Red Asistencial.

Desde la perspectiva de gestión en la Red APS y niveles hospitalarios del SS Chiloé, la regresión logística aporta interpretabilidad directa: sus coeficientes, transformados en odds ratios, identifican claramente a la edad, presión arterial sistólica e índice cintura–talla como predictores clave . Esto facilita la comunicación con equipos clínicos y la integración en protocolos de tamizaje prehospitalario y evaluación de riesgo cardiovascular en atención primaria. Por su parte, XGBoost, con su capacidad de ensamble y regularización automática, despliega un mayor poder discriminativo —reflejado en un AUC superior— y una robustez natural frente a valores faltantes y atípicos, destacando variables como IMC, PAS, colesterol total, ICT y edad . Sin embargo, su complejidad computacional y la necesidad de técnicas de explicación (por ejemplo, SHAP) exigen infraestructura y personal especializado.

6 Referencias:

- Atalah, E., et al. (2003). Prevalencia de factores de riesgo de enfermedad cardiovascular en trabajadores de empresas de servicios. Revista Médica de Chile, 131(2), 123-130. https://dx.doi.org/10.4067/S0034-98872003000200001
- Pedrero, V., et al. (2021). Generalidades del Machine Learning y su aplicación en la gestión sanitaria en Servicios de Urgencia. Revista Médica de Chile, 149(2), 248-254. https://www.researchgate.net/publication/352105918_2021_Generalidades_Machine_Learning_y_su_aplicacion_en_la_gestion_sanitaria_en_SU
- González, C., et al. (2016). Prevalencia de factores de riesgo cardiovascular en trabajadores de salud. Revista Chilena de Nutrición, 43(1), 10-16. https://dx.doi.org/10.4067/S0717-75182016000100005
- Cárdenas, Claudio, González, Sergio, Nahuel, Rosa, Herrera, Pablo, Ferrada, Luis, & Celis, Diego. (2018). Diseño de un modelo predictivo de pesquisa cardiovascular utilizando Árboles de Decisión: propensión de pacientes a presentar diabetes tipo 2, hipertensión arterial o dislipidemia: Estudio piloto, comuna de Quellón, Chiloé. Revista chilena de cardiología, 37(2), 126-133. https://dx.doi.org/10.4067/S0718-85602018000200126
- Hernández Rodríguez, José, & Duchi Jimbo, Paola Narcisa. (2015). Índice cintura/talla y su utilidad para detectar riesgo cardiovascular y metabólico. Revista Cubana de Endocrinología, 26(1), 66-76. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1561-29532015000100006&lng=es&tlng=es
- Khan, S. S., Ning, H., Wilkins, J. T., Allen, N., Carnethon, M., Berry, J. D., Sweis, R. N., & Lloyd-Jones, D. M. (2018). Association of Body Mass Index With Lifetime Risk of Cardiovascular Disease and Compression of Morbidity. JAMA cardiology, 3(4), 280–287. https://doi.org/10.1001/jamacardio.2018.0022

- https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/
- https://xgboosting.com/xgboost-advantages-and-disadvantages-pros-vs-cons/
- https://builtin.com/data-science/train-test-split
- https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/
- https://realpython.com/train-test-split-python-data/