Practica 2 – Limpieza y validación de los datos

M2.854 - Tipología de vida y Ciclo de los Datos

Juan Carlos Ghiringhelli Jueguen, Juan Pablo Botero Suaza

6 de junio, 2019

Contents

---

1 Introducción

1.1 Descripción del dataset

El conjunto de datos elegido registra, en el contexto de consultas para diagnosticar diabetes, datos predictores médicos provenientes de un conjunto de pruebas realizadas a mujeres de la India.

Como suele ocurrir con datos predictivos médicos, es posible anticipar el diagnóstico de la condición médica para pacientes sin necesidad de realizar la prueba específica. Para casos de alto riesgo se podrían tomar las medidas adecuadas como notificación al paciente o tomar precauciones ante el ingreso del paciente a una operación o una emergencia. También sirve para comprender mejor las causas de la condición y la correlación de diferentes valores con la posibilidad de padecerla.

El contenido se descargó del siguiente enlace de Kaggle: https://www.kaggle.com/uciml/pima-indians-diabetes-database y contiene nueve atributos:

1. Pregnancies: Cantidad de veces que la paciente estuvo embarazada.
2. Glucose: Concentración de glucosa en sangre dentro de dos horas de una prueba de resistencia oral a la glucosa, medido en miligramo por decilitro.
3. BloodPressure: Presión de sangre, en milímetros por mercurio, una medida médica equivalente a la presión de una columna de mercurio de un mm de alto a 0°C a una atmosfera.
4. SkinThickness: grosor de la piel en mm en la zona del pliegue del tríceps.
5. Insulin: Insulina administrada por suero en la ultima hora, en unidades por mililitro.
6. BMI: de Body Mass Index, índice de masa corporal, medido en $(\text{kilos/altura})^2$.
7. DiabetesPedigreeFunction: función de la condición presente en parientes para asignar una probabilidad genética de heredarla.
8. Age: edad en años.
9. Outcome: Presencia de la condición diabetes, siendo 1 positivo y 0 negativo. El conjunto presenta 268 casos positivos sobre un total de 768.

---

2 Análisis descriptivo

2.1 Carga de los datos

Cargar el archivo de datos diabetes.csv, validar que los tipos de datos son los correctos. Si no es así, conviertelos al tipo de datos adecuado.

```
indian_diabetes<-read.csv("diabetes.csv", header=TRUE,sep=",",dec=".")
str(indian_diabetes)
```

```
## 'data.frame':    768 obs. of  9 variables:
##  $ Pregnancies             : int  6 1 8 1 0 5 3 10 2 8 ...
##  $ Glucose                 : int  148 85 183 89 137 116 78 115 197 125 ...
##  $ BloodPressure           : int  72 66 64 66 40 74 50 0 70 96 ...
##  $ SkinThickness           : int  35 29 0 23 35 0 32 0 45 0 ...
##  $ Insulin                 : int  0 0 0 94 168 0 88 0 543 0 ...
##  $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
##  $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
##  $ Outcome                 : int  1 0 1 0 1 0 1 0 1 1 ...
```

El set de datos contiene 9 variables con un total de 768 observaciones

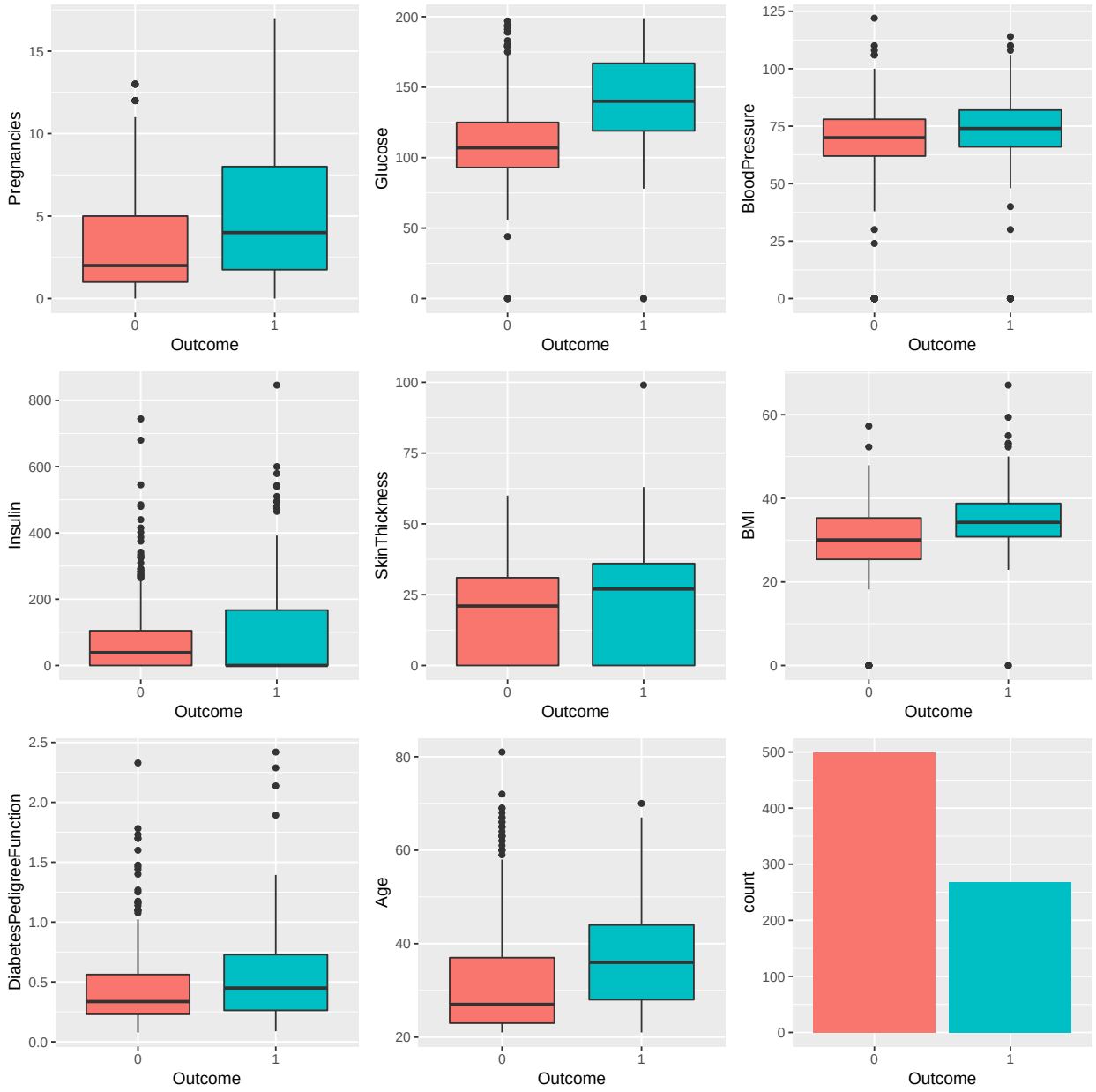
```
#Convertir variable objetivo a tipo factor para análisis posteriores
indian_diabetes$Outcome<-as.factor(indian_diabetes$Outcome)
head(indian_diabetes)
```

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0

```
summary(indian_diabetes)
```

```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##     Insulin           BMI        DiabetesPedigreeFunction      Age
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780           Min.   :21.00
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437           1st Qu.:24.00
##  Median : 30.5   Median :32.00   Median :0.3725           Median :29.00
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719           Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262           3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200           Max.   :81.00
##  Outcome
##  0:500
##  1:268
##
##
##
##
```

2.2 Análisis visual



A partir de los gráficos de boxplot podemos inferir que algunas variables pueden tener mayor incidencia sobre la condición diabética para una mujer de la muestra, donde resalta notoriamente el número de embarazos(Pregnancies), la edad (Age), el nivel de glucosa (Glucose) y el índice de masa corporal (BMI), en los apartados posteriores analizaremos si esta afirmación gráfica tambien tiene su correspondencia desde el punto de vista estadístico inferenial/predictivo.

3 Integración y selección de los datos de interés a analizar

Inicialmente, para nuestro caso se consideraran todas las variables independientes para construir el modelo explicativo sobre el atributo Outcome, en caso de identificarse variables estadísticamente no significativas durante el proceso, serán exluídas de dichos modelos con el fin de obtener un resultado más adecuado para explicar el diagnóstico de presencia de diabetes sobre esta muestra de la población.

---

4 Limpieza de los datos

4.1 Identificación y tratamiento de valores nulos, vacios y ceros

```
colSums(is.na(indian_diabetes))
```

```
##              Pregnancies                  Glucose            BloodPressure
##                        0                        0                        0
##            SkinThickness                  Insulin                      BMI
##                        0                        0                        0
## DiabetesPedigreeFunction                      Age                  Outcome
##                        0                        0                        0
```

```
colSums(indian_diabetes=="")
```

```
##              Pregnancies                  Glucose            BloodPressure
##                        0                        0                        0
##            SkinThickness                  Insulin                      BMI
##                        0                        0                        0
## DiabetesPedigreeFunction                      Age                  Outcome
##                        0                        0                        0
```

```
colSums(indian_diabetes==0)
```

```
##              Pregnancies                  Glucose            BloodPressure
##                      111                        5                       35
##            SkinThickness                  Insulin                      BMI
##                      227                      374                       11
## DiabetesPedigreeFunction                      Age                  Outcome
##                        0                        0                      500
```

El conjunto de datos no presenta campos vacíos ni nulos, por lo que no será necesario tratarlos en este aspecto.

Los atributos de cantidad de embarazos e insulina en sangre presentan casos con 0, lo cual es coherente, tanto por no haber estado embarazada nunca la persona como por no consumir insulina. Si bien puede resultar de una falta de datos, se asumirá que todos estos casos son válidos.

Los valores de glucosa, presión de sangre, grosor de la piel en el tríceps, e índice de masa corporal presentan valores en cero. Consideraremos que esto denota una falta de datos, ya que en cualquiera de estos casos el paciente estaría muerto o herido de gravedad.

Para cada caso:

• Glucosa: el nivel de glucosa en sangre, a diferencia de la presión, no tiene un valor constante que refleja un buen estado de salud. Si bien los extremos siempre son peligrosos, esta presenta normalmente una gran varianza dependiendo de la última vez que se consumió glucosa, que cantidad, en que forma, con que alimentos y dependiendo del metabolismo y actividad inmediata. Dentro de los valores analizados, es el que presenta más variabilidad y valores extremos, aunque de los 768 casos, solo 5 presentan un 0, por lo que utilizaremos el mediano. Otra opción sería, dada la baja relación entre casos con valores faltantes y casos totales, eliminar los registros totalmente.

• Presión de sangre: Muchos valores presentan una variación muy alta o baja, y ni el promedio ni la mediana representan realmente un valor neutro, lo que haría que puedan tener un peso sobre la predicción siendo un valor desconocido. Utilizando registros médicos, seleccionaremos el valor 105, valor considerado normal para mujeres de cualquier edad, dando por supuesto que si no se sabe el valor es porque no se consideró importante anotarlo o medirlo.

• Grosor de piel: Los valores presentan un rango mediano y valores bien distribuidos, y presentan el conteo más alto de valores faltantes. Se presentan 227 casos sobre el total, una cantidad importante, por lo que usaremos el método de los k vecinos más cercanos, más robusto que la mediana o promedio.

• Índice de masa corporal (IBM): similar caso al grosor de piel, pero con un número mucho menor de casos, solo 10, usaremos el promedio, al presentar los datos valores con baja varianza y simétricos.

4.2 Imputación de valores

A continuación, se ejectuaran los scripts para imputación de datos vacios.

```
#imputar glucose con valor de la mediana
glucose_median <- median(indian_diabetes$Glucose[indian_diabetes$Glucose!=0])
indian_diabetes$Glucose <- ifelse(indian_diabetes$Glucose==0, glucose_median, indian_diabetes$Glucose)
#imputar blood pressure con valor tipico
indian_diabetes$BloodPressure <- ifelse(indian_diabetes$BloodPressure==0, 105, indian_diabetes$BloodPres
#imputar skin thick con valor de la media
library(DMwR)
indian_diabetes$SkinThickness[indian_diabetes$SkinThickness== 0] <- NA
#usando parámetros por defecto, k=10, weighted average, scale = T
indian_diabetes <- knnImputation(as.data.frame(indian_diabetes))
indian_diabetes$SkinThickness<-round(indian_diabetes$SkinThickness,digits = 0)
#imputar BMI con valor de la media
bmi_mean <- lapply(mean(indian_diabetes$BMI[indian_diabetes$BMI!=0]), round, 1)[[1]]
indian_diabetes$BMI <- ifelse(indian_diabetes$BMI==0, bmi_mean, indian_diabetes$BMI)
```

4.3 Identificación y tratamiento de valores extremos

Utilizaremos *boxplots* para analizar los valores extremos. Utilizaremos el rango inter cuartil, la diferencia entre los valores de los cuartiles 75 y 25, para detectar valores extremos.

```r
#analisis outliers
outlier_glucose <- boxplot.stats(indian_diabetes$Glucose)$out
outlier_glucose
```

```
## integer(0)
```

```r
outlier_bmi <- boxplot.stats(indian_diabetes$BMI)$out
outlier_bmi
```

```
## [1] 53.2 55.0 67.1 52.3 52.3 52.9 59.4 57.3
```

```r
outlier_ins <- boxplot.stats(indian_diabetes$Insulin)$out
outlier_ins
```

```
##  [1] 543 846 342 495 325 485 495 478 744 370 680 402 375 545 360 325 465
## [18] 325 415 579 474 328 480 326 330 600 321 440 540 480 335 387 392 510
```

```r
outlier_age <- boxplot.stats(indian_diabetes$Age)$out
outlier_age
```

```
## [1] 69 67 72 81 67 67 70 68 69
```

```r
outlier_pregnancies <- boxplot.stats(indian_diabetes$Pregnancies)$out
outlier_pregnancies
```

```
## [1] 15 17 14 14
```

```r
outlier_bpressure <- boxplot.stats(indian_diabetes$BloodPressure)$out
outlier_bpressure
```

```
## [1]  30 110 122  30 110 110  24 114
```

```r
outlier_skin <- boxplot.stats(indian_diabetes$SkinThickness)$out
outlier_skin
```

```
## [1] 60 54 56 54 63 99
```

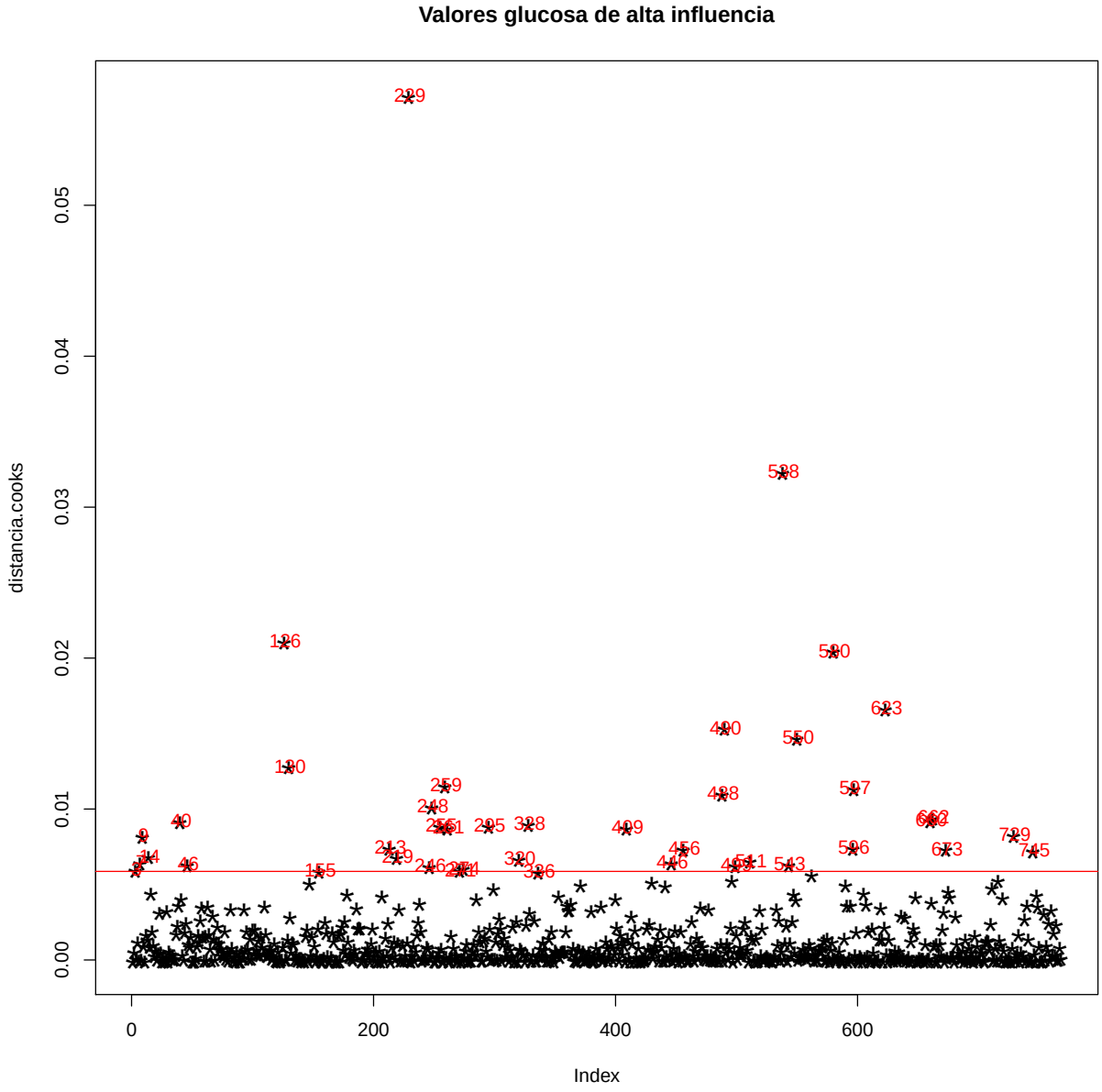
```r
outlier_dpf <- boxplot.stats(indian_diabetes$DiabetesPedigreeFunction)$out
outlier_dpf
```

```
##  [1] 2.288 1.441 1.390 1.893 1.781 1.222 1.400 1.321 1.224 2.329 1.318
## [12] 1.213 1.353 1.224 1.391 1.476 2.137 1.731 1.268 1.600 2.420 1.251
## [23] 1.699 1.258 1.282 1.698 1.461 1.292 1.394
```

Utilizaremos este paso para detectar a simple vista los casos más importantes, permitiéndonos seleccionar atributos problemáticos. Si bien la insulina presenta muchos valores extremos, este valor mide la dosis dada al paciente en la última hora, por lo cual es normal que los valores sean arbitrariamente muy diferentes. Para detectar explicitamente los valores, utilizaremos un método multi-variable. Crearemos un modelo lineal con las observaciones de la glucosa, por ser el valor con más importancia detectado anteriormente y por no presentar valores extremos en las *boxplots*, y utilizando el conjunto total de datos menos la prueba de insulina, ya que el valor 0 es muy común y el hecho de que se le haya suministrado al paciente en la última hora no debería tener gran influencia en el padecimiento de la condición.

Calcularemos la distancia de *Cooks*, y luego extraeremos los valores más influenciables, seleccionados como los valores cuya distancia de Cooks sea cuatro veces la media.

```
#modelo lineal
indian_diabetes_modelo_lineal <- indian_diabetes
indian_diabetes_modelo_lineal$Insulin <- NULL
modelo.lineal <- lm(indian_diabetes_modelo_lineal$Glucose ~ ., data=indian_diabetes_modelo_lineal)
distancia.cooks <- cooks.distance(modelo.lineal)
plot(distancia.cooks, pch="*", cex=2, main="Valores glucosa de alta influencia")
#agregar linea de corte con distancia 4 de la media
abline(h = 4*mean(distancia.cooks, na.rm=T), col="red")
text(x=1:length(distancia.cooks)+1, y=distancia.cooks, labels=ifelse(distancia.cooks>4*mean(distancia.c
```




```
#devuelve las filas con mayor influencia
influentes <- as.numeric(names(distancia.cooks)[(distancia.cooks > 4*mean(distancia.cooks, na.rm=T))])
indian_diabetes[influentes,]
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcor
3	8	183	64	29	0	23.3	0.672	32	1
7	3	78	50	32	88	31.0	0.248	26	1
9	2	197	70	45	543	30.5	0.158	53	1
14	1	189	60	23	846	30.1	0.398	59	1
40	4	111	72	47	207	37.1	1.390	56	1
46	0	180	66	39	0	42.0	1.893	25	1
126	1	88	30	42	99	55.0	0.496	26	1
130	0	105	84	27	0	27.9	0.741	62	1

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcor
155	8	188	78	35	0	47.9	0.137	43	1
213	7	179	95	31	0	34.2	0.164	60	0
219	5	85	74	22	0	29.0	1.224	32	1
229	4	197	70	39	744	36.7	2.329	31	0
246	9	184	85	15	0	30.0	1.213	49	1
248	0	165	90	33	680	52.3	0.427	23	0
255	12	92	62	7	258	27.6	0.926	44	1
259	1	193	50	16	375	25.9	0.655	24	0
261	3	191	68	15	130	30.9	0.299	34	0
271	10	101	86	37	0	45.6	1.136	38	1
274	1	71	78	50	45	33.2	0.422	21	0
295	0	161	50	35	0	21.9	0.254	65	0
320	6	194	78	33	0	23.5	0.129	59	1
328	10	179	70	31	0	35.1	0.200	37	0
336	0	165	76	43	255	47.9	0.259	26	0
409	8	197	74	27	0	25.9	1.191	39	1
446	0	180	78	63	14	59.4	2.420	25	1
456	14	175	62	30	0	33.6	0.212	38	1
488	0	173	78	32	265	46.5	1.159	58	0
490	8	194	80	36	0	26.1	0.551	67	0
499	7	195	70	33	145	25.1	0.163	55	1
511	12	84	72	31	0	29.7	0.297	46	1
538	0	57	60	23	0	21.7	0.735	67	0
543	10	90	85	32	0	34.9	0.825	56	1
550	4	189	110	31	0	28.5	0.680	37	0
580	2	197	70	99	0	34.7	0.575	62	1
596	0	188	82	14	185	32.0	0.682	22	1
597	0	67	76	35	0	45.3	0.194	46	0
623	6	183	94	28	0	40.8	1.461	45	0
660	3	80	82	31	70	34.2	1.292	27	1
662	1	199	76	43	0	42.9	1.394	22	1
673	10	68	106	23	49	35.5	0.285	47	0
729	2	175	88	24	0	22.9	0.326	22	0
745	13	153	88	37	140	40.6	1.174	39	0

La intuición marca que en general veremos valores de muy alto o muy bajo valor en alguno de los atributos, en particular la glocusa, lo que se confirma. Analizando algunos casos en particular:

• La fila 9 presenta un valor de glucosa de 197, extremadamente alto. • La fila 255 presenta valores de glucosa, presión sanguinea normal, pero 12 embarazos y un grosor de piel muy delgado, de 7. • La fila 271 oresenta un valor normal de glucosa, pero 10 embarazos y un coeficiente de pedigree de 1.136. • La fila 446 presenta un valor alto de glucosa, y un coeficiente de pedigree de 2.420, extraordinariamente alto. • La fila 538 presenta un valor muy bajo de glucosa, 57, similar a la presión sanguinea, 60.

No parece haber un caso donde la edad sea un factor preponderante, probablemente influenciado por el rango acotado de valores de la muestra.

Eliminaremos de los datos las filas para estos valores extremos. A diferencia de la imputación de valores vacios, donde se busca eliminar el impacto de estos valores al aproximarlos a un valor esperado pero pudiendo utlizar el resto de los atributos, modificar estos tiene gran potencial de disrupción en los modelos.

```
indian_diabetes <- indian_diabetes[-c(influentes), ]
write.csv(indian_diabetes,file = "clean_diabetes.csv",row.names = FALSE)
```

5 Análisis de los datos

5.1 Correlaciones

Estudiaremos la correlación entre variables, para analizar la dependencia entre ellas. En el caso de encontrar una dependencia lineal, eliminaremos un atributo, dado que no es necesario uno al tener el otro.

```
#correlacion entre variables independientes
cor_matrix <- cor(indian_diabetes[,1:8])
cor_matrix
```

```
##                           Pregnancies   Glucose BloodPressure SkinThickness
## Pregnancies                1.00000000 0.1410902     0.1713111    0.16334493
## Glucose                    0.14109018 1.0000000     0.1913844    0.24336986
## BloodPressure              0.17131107 0.1913844     1.0000000    0.18414183
## SkinThickness              0.16334493 0.2433699     0.1841418    1.00000000
## Insulin                   -0.05140675 0.3405228    -0.1125577    0.14233303
## BMI                        0.03851751 0.2564738     0.2555131    0.67507802
## DiabetesPedigreeFunction -0.03852523 0.1006781    -0.0491397    0.09467312
## Age                        0.57888325 0.2707363     0.2778321    0.15909297
##                              Insulin        BMI DiabetesPedigreeFunction
## Pregnancies               -0.05140675 0.03851751              -0.03852523
## Glucose                    0.34052281 0.25647376               0.10067814
## BloodPressure             -0.11255774 0.25551306              -0.04913970
## SkinThickness              0.14233303 0.67507802               0.09467312
## Insulin                    1.00000000 0.19112397               0.19521018
## BMI                        0.19112397 1.00000000               0.11372856
## DiabetesPedigreeFunction   0.19521018 0.11372856               1.00000000
## Age                       -0.05653129 0.05026864               0.03431833
##                                  Age
## Pregnancies               0.57888325
## Glucose                   0.27073629
## BloodPressure             0.27783207
## SkinThickness             0.15909297
## Insulin                  -0.05653129
## BMI                       0.05026864
## DiabetesPedigreeFunction  0.03431833
## Age                       1.00000000
```

Observando la tabla, la relación más fuerte es entre el número de embarazos y la edad, algo razonable, dado que ambas cosas están relacionadas directamente con el paso del tiempo. La franja de años fertilidad femenina, de cierta flexibilidad, más las propias decisiones propias de cada persona que se suman a los modernos métodos anticonceptivos, hace que la relación sea moderada, no lo suficiente para ser considera lineal.

De este análisis deducimos que todas las variables son valiosas por si mismas al presentar un nivel de indepedencia lineal bajo, por lo tanto no se excluira inicialmente ninguna variable para la construcción del modelo de predicción.

5.2 Comprobación de la normalidad.

Utilizar test Shapiro-Wilk para confirmar presunción de normalidad.

H0: los datos provienen de una distribución normal.
H1: los datos no provienen de una distribución normal.
Nivel de significancia: 0.05

```
shapiro.test(indian_diabetes$Glucose)$p.value
```

```
## [1] 2.750372e-09
```

```
shapiro.test(indian_diabetes$Age)$p.value
```

```
## [1] 5.843501e-24
```

```
shapiro.test(indian_diabetes$BloodPressure)$p.value
```

```
## [1] 3.038788e-09
```

```
shapiro.test(indian_diabetes$SkinThickness)$p.value
```

```
## [1] 0.009484577
```

```
shapiro.test(indian_diabetes$DiabetesPedigreeFunction)$p.value
```

```
## [1] 5.836721e-25
```

```
shapiro.test(indian_diabetes$Insulin)$p.value
```

```
## [1] 1.236785e-31
```

```
shapiro.test(indian_diabetes$BMI)$p.value
```

```
## [1] 2.370593e-07
```

```
shapiro.test(indian_diabetes$Pregnancies)$p.value
```

```
## [1] 7.990509e-21
```

De acuerdo al test de Shapiro-Wilk, todo los **valores p** obtenidos son inferiores al nivel de significancia, por lo tanto se rechaza la hipotesis nula, es decir ninguna de la variables proviene de una distribución normal.

5.3 Analisis inferencial y homogeneidad de la varianza

Utilizaremos el test de "Levene"" para confirmar la homogeneidad de la varianza de acuerdo a los resultados obtenidos en el test de normalidad. En este caso se utiliza un test no parámetrico dado que ninguna de las variables cumple el supuesto de normalidad.

H0: las varianzas de las muestras son iguales.
H1: el menos 2 de las varianzas difieren en las muestras.
Nivel de significancia: 0.05.

Preguntas:

1. El valor medio en la presión sanguinea (BloodPressure) es estadísticamente significativo para mujeres que presentan diabetes (Outcome=1) respecto a aquellas que no la presentan (Outcome=0) ?.

```
library(car)

leveneTest(BloodPressure ~ Outcome, data = indian_diabetes)
```

	Df	F value	Pr(>F)
group	1	0.425721	0.5143032
	724	NA	NA

```
#La prueba revela un valor de p mayor que 0.05, lo que indica que no hay una diferencia significativa e

#Test no paramétrico para variable BloodPressure
bp_sdiabetes<- indian_diabetes[indian_diabetes$Outcome==1, "BloodPressure"]
bp_ndiabetes<- indian_diabetes[indian_diabetes$Outcome==0, "BloodPressure"]

wilcox.test(bp_sdiabetes,bp_ndiabetes,alternative="greater")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  bp_sdiabetes and bp_ndiabetes
## W = 73202, p-value = 3.335e-08
## alternative hypothesis: true location shift is greater than 0
```

```
# Como el valor p resultante  es menor que el nivel de significancia de 0.05, rechazamos la hipótesis n
```

2. El valor medio en la glucosa (Glucose) es estadísticamente significativo para mujeres que presentan diabetes (Outcome=1) respecto a aquellas que no la presentan (Outcome=0) ?.

```
leveneTest(Glucose ~ Outcome, data = indian_diabetes)
```

	Df	F value	Pr(>F)
group	1	23.91609	1.2e-06
	724	NA	NA

```r
#la prueba revela un valor de p menor que 0.05, lo que indica que hay una diferencia significativa en l

#Test no paramétrico para variable Glucose
gl_sdiabetes<- indian_diabetes[indian_diabetes$Outcome==1, "Glucose"]
gl_ndiabetes<- indian_diabetes[indian_diabetes$Outcome==0, "Glucose"]

wilcox.test(gl_sdiabetes,gl_ndiabetes,alternative="greater")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  gl_sdiabetes and gl_ndiabetes
## W = 96512, p-value < 2.2e-16
## alternative hypothesis: true location shift is greater than 0
```

```r
# Como el valor p resultante  es menor que el nivel de significancia de 0.05, rechazamos la hipótesis n
```

3. El valor medio en el nivel de insulina (Insulin) es estadísticamente significativo para mujeres que presentan diabetes (Outcome=1) respecto a aquellas que no la presentan (Outcome=0) ?.

```r
leveneTest(Insulin ~ Outcome, data = indian_diabetes)
```

	Df	F value	Pr(>F)
group	1	23.73993	1.4e-06
	724	NA	NA

```r
#la prueba revela un valor de p menor que 0.05, lo que indica que hay una diferencia significativa en l

#Test no paramétrico para variable Insulin
inl_sdiabetes<- indian_diabetes[indian_diabetes$Outcome==1, "Insulin"]
inl_ndiabetes<- indian_diabetes[indian_diabetes$Outcome==0, "Insulin"]

wilcox.test(inl_sdiabetes,inl_ndiabetes)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  inl_sdiabetes and inl_ndiabetes
## W = 64256, p-value = 0.0301
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Como el valor p resultante  es menor que el nivel de significancia de 0.05, rechazamos la hipótesis n
```

4. El valor medio en la edad (Age) es estadísticamente significativo para mujeres que presentan diabetes (Outcome=1) respecto a aquellas que no la presentan (Outcome=0) ?.

```r
leveneTest(Age ~ Outcome, data = indian_diabetes)
```

	Df	F value	Pr(>F)
group	1	1.871388	0.1717406
	724	NA	NA

```
#la prueba revela un valor de p mayor que 0.05, lo que indica que no existe una diferencia significativ

#Test no paramétrico para variable Age
age_sdiabetes<- indian_diabetes[indian_diabetes$Outcome==1, "Age"]
age_ndiabetes<- indian_diabetes[indian_diabetes$Outcome==0, "Age"]

wilcox.test(age_sdiabetes,age_ndiabetes)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  age_sdiabetes and age_ndiabetes
## W = 81464, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```
# Como el valor p resultante  es menor que el nivel de significancia de 0.05, rechazamos la hipótesis n
```

5. El valor medio en en el número de embarazos (Pregnancies) es estadísticamente significativo para mujeres que presentan diabetes (Outcome=1) respecto a aquellas que no la presentan (Outcome=0) ?.

```
leveneTest(Pregnancies ~ Outcome, data = indian_diabetes)
```

	Df	F value	Pr(>F)
group	1	20.24957	7.9e-06
	724	NA	NA

```
#la prueba revela un valor de p menor que 0.05, lo que indica que existe una diferencia significativa e

#Test no paramétrico para variable Pregnancies
prg_sdiabetes<- indian_diabetes[indian_diabetes$Outcome==1, "Pregnancies"]
prg_ndiabetes<- indian_diabetes[indian_diabetes$Outcome==0, "Pregnancies"]

wilcox.test(prg_sdiabetes,prg_ndiabetes)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  prg_sdiabetes and prg_ndiabetes
## W = 73038, p-value = 7.886e-08
## alternative hypothesis: true location shift is not equal to 0
```

```
# Como el valor p resultante  es menor que el nivel de significancia de 0.05, rechazamos la hipótesis n
```

6. El valor medio en el indice de masa corporal (BMI) es estadísticamente significativo para mujeres que presentan diabetes (Outcome=1) respecto a aquellas que no la presentan (Outcome=0) ?.

```
leveneTest(BMI ~ Outcome, data = indian_diabetes)
```

	Df	F value	Pr(>F)
group	1	2.370586	0.1240776
	724	NA	NA

```
#la prueba revela un valor de p mayor que 0.05, lo que indica que no existe una diferencia significativ

#Test no paramétrico para variable BMI
bmi_sdiabetes<- indian_diabetes[indian_diabetes$Outcome==1, "BMI"]
bmi_ndiabetes<- indian_diabetes[indian_diabetes$Outcome==0, "BMI"]

wilcox.test(bmi_sdiabetes,bmi_ndiabetes)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  bmi_sdiabetes and bmi_ndiabetes
## W = 82353, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```
# Como el valor p resultante  es menor que el nivel de significancia de 0.05, rechazamos la hipótesis n
```

7. El valor medio en la variable SkinThickness es estadísticamente significativo para mujeres que presentan diabetes (Outcome=1) respecto a aquellas que no la presentan (Outcome=0) ?.

```
leveneTest(SkinThickness ~ Outcome, data = indian_diabetes)
```

	Df	F value	Pr(>F)
group	1	15.96157	7.13e-05
	724	NA	NA

```
#la prueba revela un valor de p menor que 0.05, lo que indica que existe una diferencia significativa e

#Test no paramétrico para variable SkinThickness
stn_sdiabetes<- indian_diabetes[indian_diabetes$Outcome==1, "SkinThickness"]
stn_ndiabetes<- indian_diabetes[indian_diabetes$Outcome==0, "SkinThickness"]

wilcox.test(stn_ndiabetes, stn_sdiabetes, alternative="less")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  stn_ndiabetes and stn_sdiabetes
```

```
## W = 37496, p-value = 6.865e-16
## alternative hypothesis: true location shift is less than 0
```

```
# Como el valor p resultante  es menor que el nivel de significancia de 0.05, rechazamos la hipótesis n
```

8. El valor medio en la variable DiabetesPedigreeFunction es estadísticamente significativo para mujeres que presentan diabetes (Outcome=1) respecto a aquellas que no la presentan (Outcome=0) ?.

```
leveneTest(DiabetesPedigreeFunction ~ Outcome, data = indian_diabetes)
```

	Df	F value	Pr(>F)
group	1	6.776345	0.0094265
	724	NA	NA

```
#la prueba revela un valor de p menor que 0.05, lo que indica que existe una diferencia significativa e

#Test no paramétrico para variable DiabetesPedigreeFunction
dpf_sdiabetes<- indian_diabetes[indian_diabetes$Outcome==1, "DiabetesPedigreeFunction"]
dpf_ndiabetes<- indian_diabetes[indian_diabetes$Outcome==0, "DiabetesPedigreeFunction"]

wilcox.test(dpf_ndiabetes, dpf_sdiabetes, alternative="less")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  dpf_ndiabetes and dpf_sdiabetes
## W = 46762, p-value = 3.224e-06
## alternative hypothesis: true location shift is less than 0
```

```
# Como el valor p resultante  es menor que el nivel de significancia de 0.05, rechazamos la hipótesis n
```

5.4 Análisis predictivo

Construir un modelo de clasificación que permita predecir a partir de las variables independientes objeto de estudio, si una mujer podria tener o no una condición diabética en su organismo.

Propuesta I: Regresion logística.

```
set.seed(1234)

library(rpart)
library(rattle)
library(gmodels)
library(partykit)
library(rpart.plot)

#Separar el set de datos en 2 muestras, una para construcción y entrenamiento del modelo, y otra para m

train_idx <- sample(1:nrow(indian_diabetes),nrow(indian_diabetes)*0.7,replace=FALSE)
```

```r
train<-indian_diabetes[train_idx,]
test<-indian_diabetes[-train_idx,]


#Modelo de regresion lineal logístico funcion glm
modelo_rl<-glm(Outcome ~  . , data=train,family=binomial(link=logit))

#Evaluar si alguno de los regresores tiene influencia significativa (p-valor del contraste individual i
summary(modelo_rl)
```

```
##
## Call:
## glm(formula = Outcome ~ ., family = binomial(link = logit), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2198  -0.6273  -0.3054   0.5406   2.2979
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -12.049726   1.172442 -10.277  < 2e-16 ***
## Pregnancies               0.147638   0.044671   3.305  0.00095 ***
## Glucose                   0.051531   0.005816   8.861  < 2e-16 ***
## BloodPressure             0.014942   0.010172   1.469  0.14185
## SkinThickness             0.017790   0.019199   0.927  0.35415
## Insulin                  -0.001541   0.001209  -1.274  0.20255
## BMI                       0.074033   0.026104   2.836  0.00457 **
## DiabetesPedigreeFunction  0.833827   0.401246   2.078  0.03770 *
## Age                       0.001285   0.013227   0.097  0.92260
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 654.27  on 507  degrees of freedom
## Residual deviance: 426.32  on 499  degrees of freedom
## AIC: 444.32
##
## Number of Fisher Scoring iterations: 5
```

```r
#Modelo de regresion con variables significativas
modelo_rl2<-glm(Outcome ~  Pregnancies+Glucose+ BMI+DiabetesPedigreeFunction , data=train,family=binomia
summary(modelo_rl2)
```

```
##
## Call:
## glm(formula = Outcome ~ Pregnancies + Glucose + BMI + DiabetesPedigreeFunction,
##     family = binomial(link = logit), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2616  -0.6518  -0.3307   0.5611   2.3105
##
```

```
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -11.029382   1.005614 -10.968  < 2e-16 ***
## Pregnancies                 0.172954   0.036264   4.769 1.85e-06 ***
## Glucose                     0.050675   0.005373   9.431  < 2e-16 ***
## BMI                         0.091656   0.019832   4.622 3.81e-06 ***
## DiabetesPedigreeFunction    0.769058   0.391723   1.963   0.0496 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 654.27  on 507  degrees of freedom
## Residual deviance: 432.51  on 503  degrees of freedom
## AIC: 442.51
##
## Number of Fisher Scoring iterations: 5
```

```
#El menor valor para el indicador AIC corresponde al modelo con los regresores *Pregnancies + Glucose +


#Matriz de confusión y precisión del modelo 1
precdb <- predict(modelo_rl,newdata=test,type='response')
precdb <- ifelse(precdb > 0.70,1,0)
misClasificError <- mean(precdb != test$Outcome)

#Al establecer el parámetro  type='response', R generará probabilidades de la forma de P (y = 1 | X). N

print(paste('Precisión en la clasificación:',round((1-misClasificError)*100,digits = 2),'%'))
```

```
## [1] "Precisión en la clasificación: 80.73 %"
```

```
CrossTable(test[,"Outcome"],precdb,prop.chisq  = FALSE, prop.c = FALSE, prop.r =FALSE,dnn = c('Reality'
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  218
##
##
##              | Prediction
##      Reality |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |       142 |         7 |       149 |
##              |     0.651 |     0.032 |           |
## -------------|-----------|-----------|-----------|
##            1 |        35 |        34 |        69 |
```

```
##              |     0.161 |     0.156 |           |
## -------------|-----------|-----------|-----------|
## Column Total |       177 |        41 |       218 |
## -------------|-----------|-----------|-----------|
##
##
```

```
#Matriz de confusión y precisión del modelo 2
precdb2 <- predict(modelo_rl2,newdata=test[,c("Pregnancies","Glucose","BMI","DiabetesPedigreeFunction")]
precdb2 <- ifelse(precdb2 > 0.70,1,0)
misClasificError2 <- mean(precdb2 != test$Outcome)

print(paste('Precisión en la clasificación:',round((1-misClasificError2)*100,digits = 2),'%'))
```

```
## [1] "Precisión en la clasificación: 81.65 %"
```

```
CrossTable(test[,"Outcome"], precdb2,prop.chisq  = FALSE, prop.c = FALSE, prop.r =FALSE,dnn = c('Reality
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  218
##
##
##              | Prediction
##      Reality |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |       144 |         5 |       149 |
##              |     0.661 |     0.023 |           |
## -------------|-----------|-----------|-----------|
##            1 |        35 |        34 |        69 |
##              |     0.161 |     0.156 |           |
## -------------|-----------|-----------|-----------|
## Column Total |       179 |        39 |       218 |
## -------------|-----------|-----------|-----------|
##
##
```

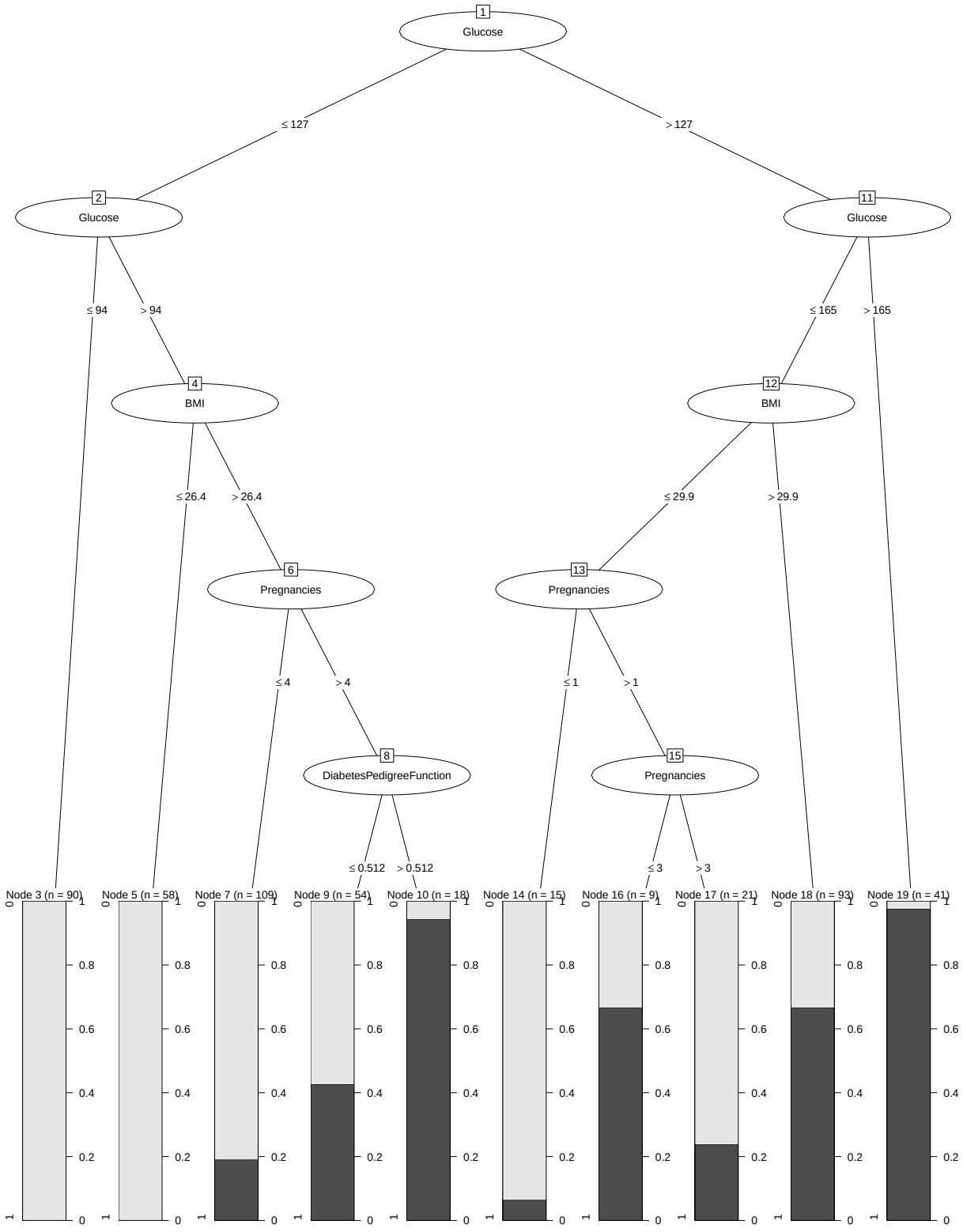
Los falsos positivos corresponde a casos en que la predicción de la probabilidad de la respuesta afirmativa es elevada , pero la respuesta observada es negativa, en nuestro caso para 9 mujeres, el modelo indica que tiene condición diabética “1”, pero en realidad no la tiene “0”.

Los falsos negativos corresponde a casos donde el modelo predice que una mujer tiene una probabilidad de condición diabética baja, sin embargo las mujeres observadas si presentan una condición diabética, 46 individuos en este caso.

Ademas se puede evidenciar que para obtener un nivel de precisión similar al modelo con todas las variables, tan solo basta con utilizar las 4 variables explicativas comentadas anteriormente.

Propuesta II: Arboles de desición C50.

```
#Ejecutar algoritmo C50 y visualizar reglas de clasificación
model.c50 <- C50::C5.0(train[,c("Pregnancies","Glucose","BMI","DiabetesPedigreeFunction")], train[,"Out
plot(model.c50)
```


1
Glucose
≤ 127
> 127
2
Glucose
11
Glucose
≤ 94
> 94
≤ 165
> 165
4
BMI
12
BMI
≤ 26.4
> 26.4
≤ 29.9
> 29.9
6
Pregnancies
13
Pregnancies
≤ 4
> 4
≤ 1
> 1
8
DiabetesPedigreeFunction
15
Pregnancies
≤ 0.512
> 0.512
≤ 3
> 3
Node 3 (n = 90)
Node 5 (n = 58)
Node 7 (n = 109)
Node 9 (n = 54)
Node 10 (n = 18)
Node 14 (n = 15)
Node 16 (n = 9)
Node 17 (n = 21)
Node 18 (n = 93)
Node 19 (n = 41)
0
0.2
0.4
0.6
0.8
1

```r
#Predicción sobre el set de datos de prueba
modelc50.predict <- predict( model.c50, test[,c("Pregnancies","Glucose","BMI","DiabetesPedigreeFunction"
print(sprintf("La precisión del árbol es: %.4f %%",100*sum(modelc50.predict == test[,"Outcome"]) / leng
```

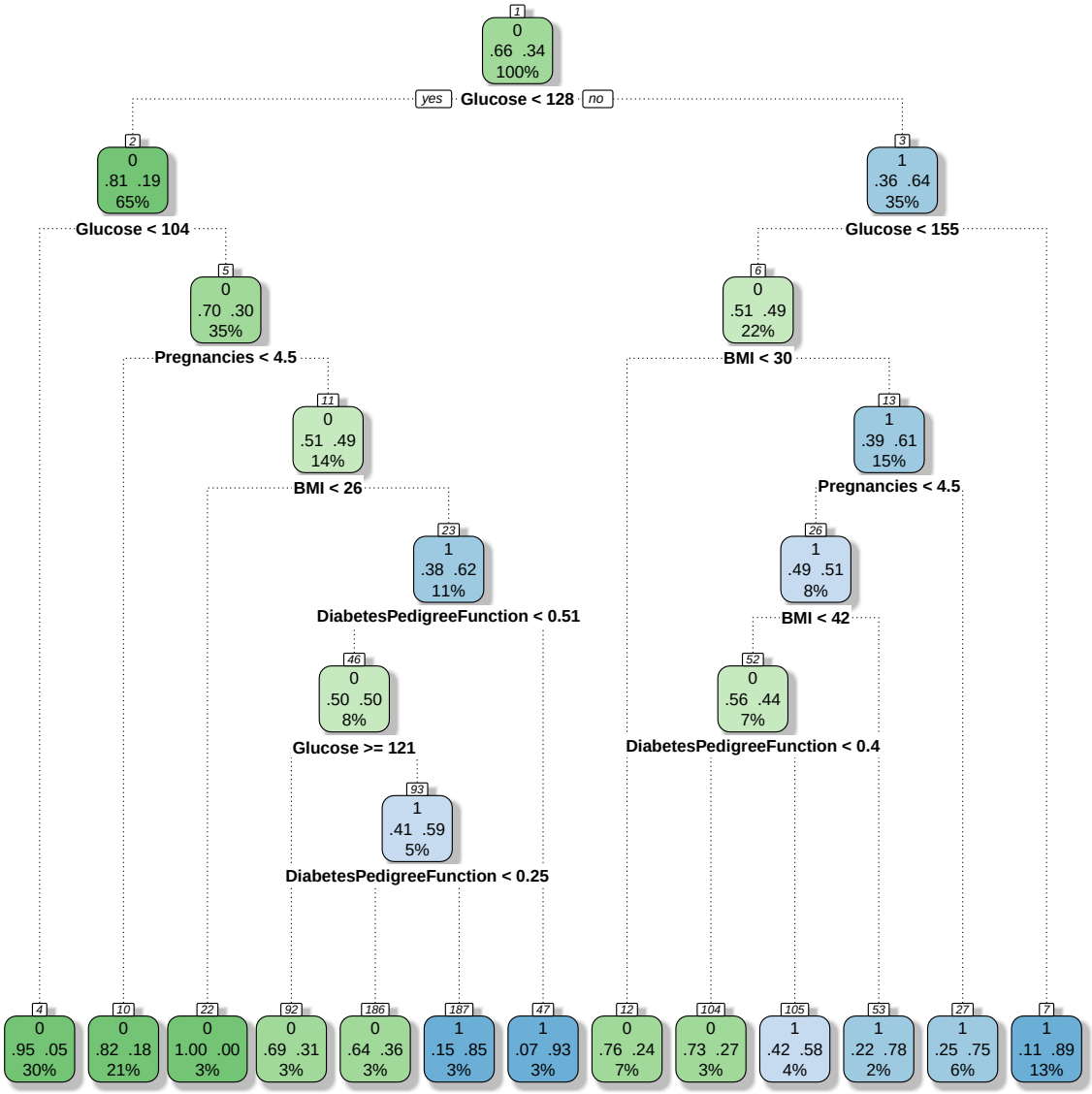
```
## [1] "La precisión del árbol es: 76.6055 %"
```

```r
#Matriz confusión
CrossTable(test[,"Outcome"], modelc50.predict,prop.chisq  = FALSE, prop.c = FALSE, prop.r =FALSE,dnn = 
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  218
##
##
##              | Prediction
##      Reality |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |       121 |        28 |       149 |
##              |     0.555 |     0.128 |           |
## -------------|-----------|-----------|-----------|
##            1 |        23 |        46 |        69 |
##              |     0.106 |     0.211 |           |
## -------------|-----------|-----------|-----------|
## Column Total |       144 |        74 |       218 |
## -------------|-----------|-----------|-----------|
##
##
```

Propuesta III: Arboles de desición CART.

```r
#Ejecutar algoritmo clasificación CART
model.cart <- rpart(Outcome~.,data=train[,c("Pregnancies","Glucose","BMI","DiabetesPedigreeFunction","Ou
fancyRpartPlot(model.cart)
```



Rattle 2019−jun.−06 20:14:34 User

```
#Predicción sobre el set de datos de prueba
modelcart.predict <- predict( model.cart, test[,c("Pregnancies","Glucose","BMI","DiabetesPedigreeFunctio
print(sprintf("La precisión del árbol es: %.4f %%",100*sum(modelcart.predict == test[,"Outcome"]) / leng
```

```
## [1] "La precisión del árbol es: 80.2752 %"
```

```
#Matriz confusión
CrossTable(test[,"Outcome"], modelcart.predict ,prop.chisq  = FALSE, prop.c = FALSE, prop.r =FALSE,dnn =
```

```
##
##
##    Cell Contents
```

```
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  218
##
##
##              | Prediction
##      Reality |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |       130 |        19 |       149 |
##              |     0.596 |     0.087 |           |
## -------------|-----------|-----------|-----------|
##            1 |        24 |        45 |        69 |
##              |     0.110 |     0.206 |           |
## -------------|-----------|-----------|-----------|
## Column Total |       154 |        64 |       218 |
## -------------|-----------|-----------|-----------|
##
##
```

---

6 Presentacion de resultados

```
#Resumen calidad de los modelos predictivos

data.frame(Modelo=c("Regresión Logística","Arboles de Decision C50","Arboles de Decisión CART"),"Precis
```

Modelo	Precisión
Regresión Logística	81.65 %
Arboles de Decision C50	76.61 %
Arboles de Decisión CART	80.28 %

```
#Predicción sobre un perfil de prueba

data_fit1<- data.frame(Pregnancies=0,Glucose=169,BMI=27.97,DiabetesPedigreeFunction=1)
data_fit1
```

Pregnancies	Glucose	BMI	DiabetesPedigreeFunction
0	169	27.97	1

```r
#Predicción GLM Binomial
modelo_rl.predict <- predict( modelo_rl2, data_fit1, type="response" )
modelo_rl.predict <- ifelse(modelo_rl.predict > 0.70,1,0)


#Predicción CART
modelcart.predict <- predict( model.cart, data_fit1, type="class")


#Predicción C50
modelc50.predict <- predict( model.c50, data_fit1, type="class" )


data.frame(Modelo=c("Regresión Logística","Arboles de Decision C50","Arboles de Decisión CART"),"Outcome
```

Modelo	Outcome
Regresión Logística	1
Arboles de Decision C50	1
Arboles de Decisión CART	1

```r
#Mismo perfil de prueba con variación en variable Glucose

data_fit2<- data.frame(Pregnancies=0,Glucose=110,BMI=27.97,DiabetesPedigreeFunction=1)
data_fit2
```

Pregnancies	Glucose	BMI	DiabetesPedigreeFunction
0	110	27.97	1

```r
#Predicción GLM Binomial
modelo_rl.predict <- predict( modelo_rl2, data_fit2, type="response" )
modelo_rl.predict <- ifelse(modelo_rl.predict > 0.70,1,0)


#Predicción CART
modelcart.predict <- predict( model.cart, data_fit2, type="class")


#Predicción C50
modelc50.predict <- predict( model.c50, data_fit2, type="class" )


data.frame(Modelo=c("Regresión Logística","Arboles de Decision C50","Arboles de Decisión CART"),"Outcome
```

Modelo	Outcome
Regresión Logística	0
Arboles de Decision C50	0
Arboles de Decisión CART	0

7 Conclusiones

• Inicialmente se ha realizado un estudio de los datos y los atributos para comprender semántica y sintácticamente el conjunto a estudiar. Posteriormente los datos fueron sometidos a preprocesamiento para imputar los ceros que carecían de sentido, análisis de valores extremos para eliminarlos, y correlación de variables para considerar la eliminación de alguna, lo que finalmente no ocurrió.

• Los análisis de correlación y de contraste de hipótesis permite analizar cuáles de los atributos estudiados ejercen una mayor influencia sobre la posibilidad de padecer diabetes, y el modelo de regresión lineal obtenido permite realizar predicciones para el diagnóstico dados otros valores conocidos que son más simples de obtener.

• Se han realizado tres tipos de pruebas estadísticas sobre el conjunto de datos relativo a observaciones médicas para mujeres de la India, ante pruebas de diabetes, con el motivo de cumplir con el objetivo que se planteaba al comienzo. Para cada una de las pruebas, se han graficado e impreso los resultados que arrojan, para extraer información valiosa referida al conjunto o una parte de él. El modelo de regresión logística, en términos de la calidad del proceso de clasificación, es el mejor de los 3 escenarios evaluados.

• A partir de los resultados obtenedios en los modelos de árboles de decisión, es posible identificar que la variable Glucose tiene gran incidencia sobre las reglas de clasificación obtenidas, podriamos decir que es el atributo más influyente al momento de determinar la presencia de diabetes sobre un individuo bajo este contexto, incluso como se ilustra en el perfil de prueba, solo la variación en la variable Glucose genera diferentes valores para la variable objetivo Outcome.

8 Bibliografía

Calvo M., Subirats L., Perez D. (2019). Introduccion a la limpieza y analisis de los datos.Editorial UOC.

Megan Squire (2015). Clean Data. Packt Publishing Ltd.

Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques.

Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369.

Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media.

Wes McKinney (2012). Python for Data Analysis. O'Reilley Media, Inc.

Tutorial de Github https://guides.github.com/activities/hello-world.