Clasificador Binario de Consolidaciones

Por Eugenia Berrino

17/5/2021

Importación de Datos

```
data <- read.csv('final_results_v1.csv', header = TRUE, sep = ',')
names(data)[3] <- "bb_cmax"
names(data)[4] <- "bb_cmean"
names(data)[5] <- "bb_cmedian"
data$class_name = as.factor(data$class_name)
pander(summary(data))
```

Table 1: Table continues below

X	bb_counts	bb_cmax	bb_cmean
Length:1385	Min. : 1.00	Min. :0.00108	Min. :0.001080
Class :character	1st Qu.: 7.00	1st Qu.:0.01004	1st Qu.:0.003607
Mode :character	Median : 11.00	Median :0.03873	Median :0.007805
NA	Mean : 14.61	Mean :0.18721	Mean :0.018467
NA	3rd Qu.: 17.00	3rd Qu.:0.28980	3rd Qu.:0.027006
NA	Max. :223.00	Max. :0.88574	Max. :0.166762

bb_cmedian	image_id	class_name
Min. :0.001072	Length:1385	Consolidation: 278
1st Qu.:0.001893	Class :character	No finding :1107
Median :0.002459	Mode :character	NA
Mean :0.002819	NA	NA
3rd Qu.:0.003265	NA	NA
Max. :0.020599	NA	NA

Modelos de Regression Logistica

Resulta importante comenzar aclarando que en el contexto de este modelo al hacer referencia al train set, estamos hablando del “extended test set” ya procesado por el algoritmo de detección de objetos YOLO, mientras que el test set es el conjunto de imágenes “de validación” dadas por el HIBA.

A continuación, se definen los modelos en base a los datos de test extendido del dataset VinBigData. El enfoque definido para determinar la presencia o ausencia de consolidaciones en imágenes a partir de los

resultados de nuestro algoritmo de detección de objetos es el siguiente: Se seleccionan como variables de salida la cantidad de bounding boxes detectadas para cada una de las imágenes y un valor de confianza "resumen" obtenido a partir de los valores de confianza de los bounding boxes de esas imágenes. Con dichas variables, se ajustan los coeficientes correspondientes a dichos modelos y finalmente se comparan los diferentes modelos con el objetivo de escoger uno de ellos para realizar la prueba final en las imágenes del Hospital.

El tipo de modelo a utilizarse es de regresión logística debido a que contamos con una variable dependiente dicotómica. Este tipo de modelos son muy similares a las regresiones lineales, con la diferencia que la variable de salida es transformada para representar una probabilidad de pertenencia a una determinada clase. Dicho objetivo se consigue gracias al uso de la función link sigmoidea, la cual se presenta a continuación:

$$\sigma(x) = \frac{1}{1 + e^{(-x)}} \tag{1}$$

Modelo 1

El primer modelo propuesto es el siguiente:

$$logit(P) = \ln(\frac{P}{1-P}) = \alpha_0 + \alpha_1 * BB_{counts} + \alpha_2 * BB_{cmax} \tag{2}$$

donde, $logit(P)$ es el logaritmo natural del ODDS del evento, α_i son los coeficientes que van a ser calculados mediante el método de máxima verosimilitud de manera de optimizar la predicción correcta, y BB_{counts} y BB_{cmax} representan las variables de nuestro modelo.

El método de máxima verosimilitud se basa en la idea de que la muestra obtenida, por haber ocurrido tiene una alta probabilidad de ocurrir y estima los parámetros como aquellos que maximizan la probabilidad de obtener nuestra muestra.Esto se logra maximizando la función de verosimilitud, que es la función de probabilidad conjunta de la muestra. Este proceso que se realiza de manera iterativa.

BB_{counts} representa el número de bounding boxes que el modelo de detección de objetos encontró para una dada imagen y BB_{cmax} la confianza máxima detectada para bounding box en dicha imagen. El motivo por el cual se consideró incluir esta última variable y considerarla para el primer modelo se fundamenta en el hecho de que es suficiente poseer una bounding box que el algoritmo determina con un elevado valor de confianza para que dicha imagen ya sea clasificada como perteneciente a la clase de consolidación.

α_0 representa la ordenada al origen, α_1 es el coeficiente que acompaña a la variables BB_{counts} y por lo tanto, un aumento unitario en la cantidad de cajas que detecta el algoritmo implica un aumento equivalente al valor de este coeficiente en $logit(P)$. Dicho comportamiento es análogo para α_2 y BB_{cmax}

Previo al entrenamiento del modelo, definimos la variable "consolidation" para garantizar que el modelo tome como outcomes positivos los pertenecientes a dicha clase y como negativos a los pertenecientes a No Finding.

```
data$consolidation[data$class_name == "Consolidation"] <- 1
data$consolidation[data$class_name == "No finding"] <- 0
```

Ahora si, corremos el modelo.

```r
m1 <- glm(consolidation ~ bb_counts + bb_cmax , data = data, family = "binomial")
summary(m1)
```

```
Call:
glm(formula = consolidation ~ bb_counts + bb_cmax, family = "binomial",
    data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7573  -0.2984  -0.1980  -0.1340   2.9862

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.15022    0.27354  -18.83   <2e-16 ***
bb_counts    0.13448    0.01279   10.51   <2e-16 ***
bb_cmax      4.81532    0.39014   12.34   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1388.88  on 1384  degrees of freedom
Residual deviance:  620.86  on 1382  degrees of freedom
AIC: 626.86

Number of Fisher Scoring iterations: 6
```

Podemos ver en los resultados de la regresión logística, que el modelo arroja una gran variedad de parámetros. El primero de ellos que resulta interesante de analizar es el P-valor que acompaña a cada una de las variables. El hecho de encontrar un p-valor implica que se realizó un test de hipótesis. En este caso, el test de hipótesis es conocido como test de Wald, y plantea como hipótesis las siguientes:

$H_0 : \alpha_i = 0$

$H_1 : \alpha_i \neq 0$

Es decir, que para cada una de las variables $alpha_i$ se realizó un test de hipótesis separado en el que se asumió que su valor era cero, se calculó el estádístico del test y se obtuvo la probabilidad de que dicho valor sea obtenido producto del azar dado que la hipótesis nula es verdadera. El resultado de dicha probabilidad es tan pequeño que nos permite concluir con un nivel de significancia superior a 0.001 que los valores de cada una de dichas variables son diferentes a cero. Por lo tanto, se debe rechazar la hipótesis nula y por ende aceptar la hipótesis alternativa.

Otro parámetro que resulta interesante tener en cuenta a la hora de analizar el modelo de regresión logística son los valores de los α_i, devueltos por el modelo como "Estimate". Tal como se explicó anteriormente, dichos parámetros representan la variación en la logit(P) para una variación unitaria de la variable del modelo que acompañan. Del modelo podemos concluir que por cada bounding box que detecta, el ODDS ratio

A su vez, el modelo devuelve los residuos que son la diferencia entre los valores de nuestro train set y las predicciones que arroja el modelo. Por lo tanto, un resultado negativo implica una sobreestimación de la variable dependiente, mientras que uno positivo una subestimación. En este caso en particular, podemos ver como hasta el tercer cuartil es negativo. Sin embargo, debemos tener cuidado para no saltar a conclusiones

al respecto, debido a que el 80% de nuestro train set son imágenes No finding, por lo que su valor es negativo. CHARLAR CON CANDE. (los residuos no se encuentran en el rango 1-0).

Modelo 2

$$logit(P) = \alpha_0 + \alpha_1 * BB_{counts} + \alpha_2 * BB_{cmean} \tag{3}$$

```
m2 <- glm(consolidation ~ bb_counts + bb_cmean , data = data, family = "binomial")
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(m2)
```

```
Call:
glm(formula = consolidation ~ bb_counts + bb_cmean, family = "binomial",
    data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6278  -0.3344  -0.2016  -0.1259   3.0196

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.57904    0.28952  -19.27   <2e-16 ***
bb_counts    0.17470    0.01288   13.56   <2e-16 ***
bb_cmean    43.12933    4.06729   10.60   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1388.9  on 1384  degrees of freedom
Residual deviance:  660.9  on 1382  degrees of freedom
AIC: 666.9

Number of Fisher Scoring iterations: 6
```

Modelo 3

$$logit(P) = \alpha_0 + \alpha_1 * BB_{counts} + \alpha_2 * BB_{cmedian} \tag{4}$$

```
m3 <- glm(consolidation ~ bb_counts + bb_cmedian , data = data, family = "binomial")
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(m3)
```

```
Call:
glm(formula = consolidation ~ bb_counts + bb_cmedian, family = "binomial",
    data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8482  -0.4296  -0.2731  -0.1651   2.7261

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -5.46622    0.29657 -18.432  < 2e-16 ***
bb_counts     0.19430    0.01215  15.996  < 2e-16 ***
bb_cmedian  257.17455   50.45257   5.097 3.44e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1388.88  on 1384  degrees of freedom
Residual deviance:  773.03  on 1382  degrees of freedom
AIC: 779.03

Number of Fisher Scoring iterations: 6
```

Modelo 4

$$logit(P) = \alpha_0 + \alpha_1 * BB_{counts} + \alpha_2 * BB_{cmax} + \alpha_3 * BB_{cmax} * BB_{counts} \tag{5}$$

```
m4 <- glm(consolidation ~ bb_counts*bb_cmax , data = data, family = "binomial")
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(m4)
```

```
Call:
glm(formula = consolidation ~ bb_counts * bb_cmax, family = "binomial",
    data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.2783  -0.3032  -0.2260  -0.1726   2.8409

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       -4.49724    0.33292 -13.509  < 2e-16 ***
bb_counts          0.09208    0.01907   4.829 1.37e-06 ***
bb_cmax            2.22636    1.00682   2.211  0.02702 *
bb_counts:bb_cmax  0.15319    0.05625   2.723  0.00646 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 1388.88  on 1384  degrees of freedom
Residual deviance:  612.45  on 1381  degrees of freedom
AIC: 620.45

Number of Fisher Scoring iterations: 7
```

Modelo 5

$$logit(P) = \alpha_0 + \alpha_1 * BB_{counts} \tag{6}$$

```
m5 <- glm(consolidation ~ bb_counts, data = data, family = "binomial")
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(m5)
```

```
Call:
glm(formula = consolidation ~ bb_counts, family = "binomial",
    data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7956  -0.4244  -0.2907  -0.1796   2.7395

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.70509    0.23119  -20.35   <2e-16 ***
bb_counts    0.19528    0.01202   16.25   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1388.9  on 1384  degrees of freedom
Residual deviance:  794.3  on 1383  degrees of freedom
AIC: 798.3

Number of Fisher Scoring iterations: 6
```

Modelo 6

$$logit(P) = \alpha_0 + \alpha_1 * BB_{cmax} \tag{7}$$

```
m6 <- glm(consolidation ~ bb_cmax , data = data, family = "binomial")
summary(m6)
```

```
Call:
glm(formula = consolidation ~ bb_cmax, family = "binomial", data = data)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2705  -0.3286  -0.2776  -0.2681   2.5838

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -3.3263     0.1533  -21.69   <2e-16 ***
bb_cmax       6.7245     0.3446   19.51   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1388.88  on 1384  degrees of freedom
Residual deviance:  780.23  on 1383  degrees of freedom
AIC: 784.23

Number of Fisher Scoring iterations: 5
```

Comparación de modelos

Para comparar los diferentes modelos, se utilizó el valor arrojado por cada uno bajo el nombre de AIC. El AIC, o Akaike's Information Criterion o Criterio de Información de Akaike, se calcula de la siguiente manera:

$$AIC = 2\ln(\frac{e^k}{L}) \tag{8}$$

Donde k representa la cantidad de parámetros del modelo, mientras que L la función de máxima verosimilitud ya optimizada.

Es decir, el criterio toma en cuenta tanto la cantidad de variables incluidas en el modelo, penalizando modelos más complejos, así como tambien la bondad de ajuste del modelo, favoreciendo a los modelos que mejor ajuste producen. Esto permite escoger los mejores modelos, evitando el *overfitting*.

Con la anterior información y teniendo en cuenta la expresión 8, se puede concluir que un menor AIC se corresponde con un mejor modelo.

Además de utilizar el AIC para escoger el mejor modelo, se calculo el área debajo de la curva ROC para diferentes puntos de corte para cada uno de los modelos. Los resultados obtenidos en ambos casos se muestran a continuación:

```
prob=predict(m1,type=c("response"))
data$prob = prob
roc1 <- roc(consolidation ~ prob, data = data)
```

```
Setting levels: control = 0, case = 1
```

```
Setting direction: controls < cases
```

```
prob=predict(m2,type=c("response"))
data$prob = prob
roc2 <- roc(consolidation ~ prob, data = data)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```
prob=predict(m3,type=c("response"))
data$prob = prob
roc3 <- roc(consolidation ~ prob, data = data)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```
prob=predict(m4,type=c("response"))
data$prob = prob
roc4 <- roc(consolidation ~ prob, data = data)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```
prob=predict(m5,type=c("response"))
data$prob = prob
roc5 <- roc(consolidation ~ prob, data = data)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```
prob=predict(m6,type=c("response"))
data$prob = prob
roc6 <- roc(consolidation ~ prob, data = data)
```

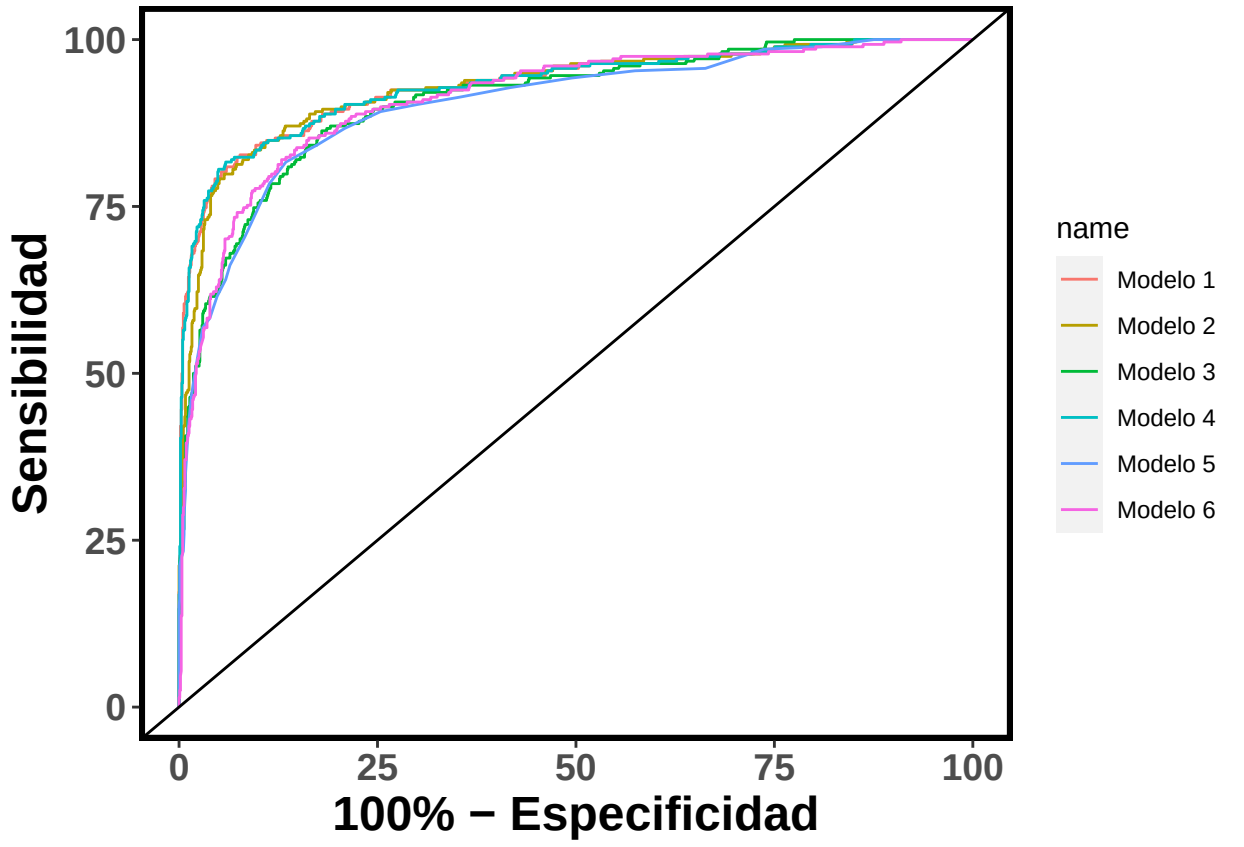
```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```
tab <- matrix(c(roc1$auc, roc2$auc, roc3$auc,
                roc4$auc, roc5$auc, roc6$auc,
                m1$aic, m2$aic, m3$aic,
                m4$aic,m5$aic,m6$aic), ncol=2, byrow=FALSE)
colnames(tab) <- c('AUC','AIC')
rownames(tab) <- c('m1','m2','m3','m4','m5','m6')
tab <- as.table(tab)
formattable(tab,
            align = "r")# INVESTIGAR COMO HACER LINDAS TABLAS CON ESTO
```

```
   AUC         AIC
m1   0.9322461 626.8574317
m2   0.9298902 666.8961848
m3   0.9092645 779.0307131
m4   0.9323176 620.4537939
m5   0.9027412 798.2960424
m6   0.9120963 784.2337671
```

```
ggroc(list("Modelo 1" = roc1, "Modelo 2" = roc2,
           "Modelo 3" = roc3, "Modelo 4" = roc4,
           "Modelo 5" = roc5, "Modelo 6" = roc6), legacy.axes = T) +
geom_abline(slope = 1 ,intercept = 0) + # add identity line
theme(
panel.background = element_blank(),
axis.title.x = element_text(size =18, face = 'bold'),
axis.title.y = element_text(size =18, face = 'bold'),
panel.border = element_rect(size = 2, fill = NA),
axis.text.x = element_text(size = 14, face ='bold'),
axis.text.y = element_text(size = 14, face ='bold')) +
xlab('100% - Especificidad') +
ylab('Sensibilidad') +
scale_x_continuous(breaks = seq(0,1,0.25), labels = seq(0,1,0.25) * 100) +
scale_y_continuous(breaks = seq(0,1,0.25), labels = seq(0,1,0.25) * 100)
```

```
Scale for 'x' is already present. Adding another scale for 'x', which will
replace the existing scale.
```



Debido a los resultados resumidos en la tabla anterior, a que la diferencia entre los dos mejores modelos es ínfima, se escogió el modelo m1.

Clasificación Test set

```
data_h <- read.csv('final_results_hiba.csv', header = TRUE, sep = ',')
names(data_h)[3] <- "bb_cmax"
names(data_h)[4] <- "bb_cmean"
names(data_h)[5] <- "bb_cmedian"
data_h$class = as.factor(data_h$class)
summary(data_h)
```

```
  file_name           bb_counts         bb_cmax            bb_cmean
 Length:1284        Min.   :  1.00   Min.   :0.001213   Min.   :0.001206
 Class :character   1st Qu.:  7.00   1st Qu.:0.008926   1st Qu.:0.003141
 Mode  :character   Median : 12.00   Median :0.026062   Median :0.005429
                    Mean   : 20.41   Mean   :0.162966   Mean   :0.015197
                    3rd Qu.: 18.00   3rd Qu.:0.167847   3rd Qu.:0.016859
                    Max.   :300.00   Max.   :0.923340   Max.   :0.272930
   bb_cmedian       file_name.1                  class
 Min.   :0.001109   Length:1284        Consolidation: 187
 1st Qu.:0.001782   Class :character   No finding   :1097
 Median :0.002200   Mode  :character
 Mean   :0.002889
 3rd Qu.:0.002985
 Max.   :0.137337
```

```
data_h$consolidation[data_h$class == "Consolidation"] <- 1
data_h$consolidation[data_h$class == "No finding"] <- 0
prob=predict(m1,newdata= data_h,type = "response")
data_h$prob = prob
write.csv(x=data_h, file="prediccion_binaria.csv")
roc1 <- roc(consolidation ~ prob,data = data_h)
```

```
Setting levels: control = 0, case = 1
```

```
Setting direction: controls < cases
```

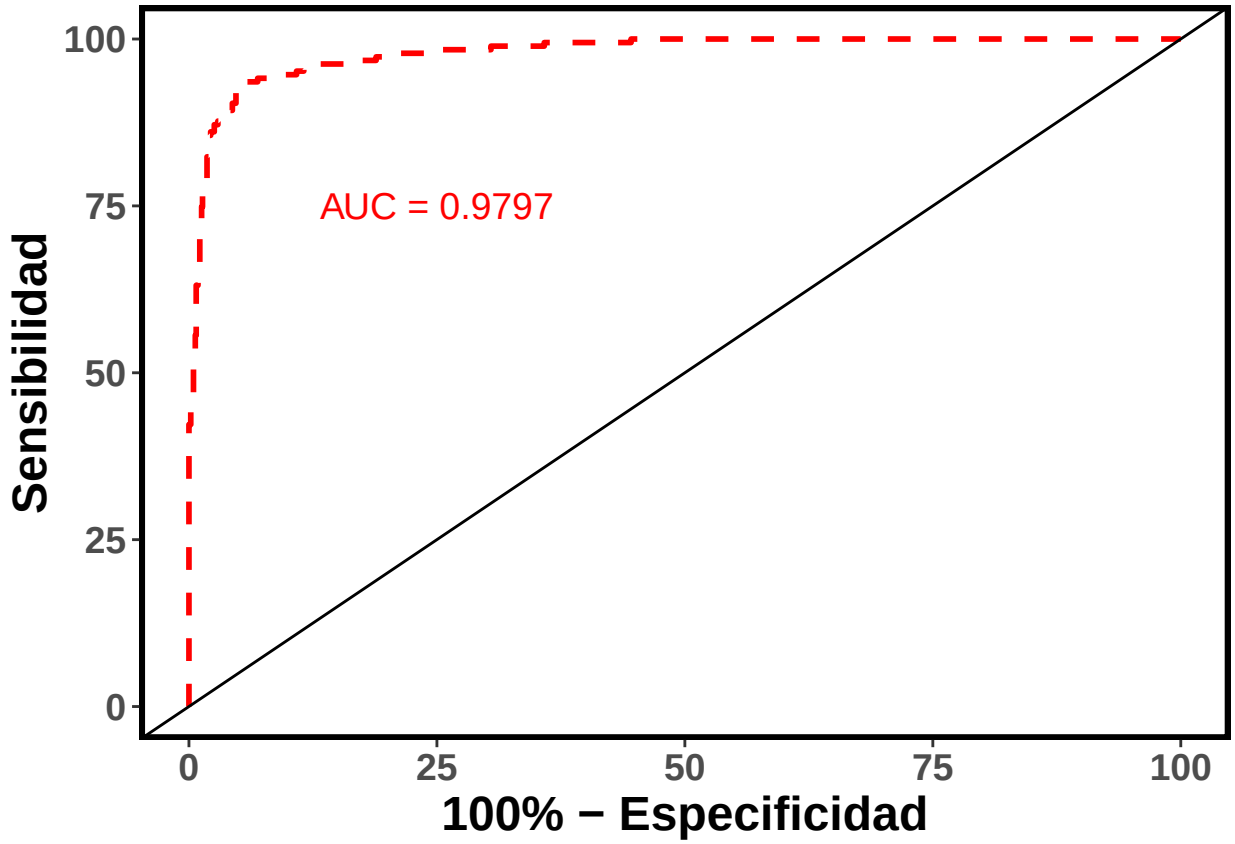
```
roc1$auc
```

```
Area under the curve: 0.9797
```

```
ggroc(roc1,alpha = 1, colour = "red",
      linetype = 2, size = 1, legacy.axes = T) +
geom_abline(slope = 1 ,intercept = 0) + # add identity line
theme(
panel.background = element_blank(),
axis.title.x = element_text(size =18, face = 'bold'),
axis.title.y = element_text(size =18, face = 'bold'),
panel.border = element_rect(size = 2, fill = NA),
axis.text.x = element_text(size = 14, face ='bold'),
axis.text.y = element_text(size = 14, face ='bold')) +
xlab('100% - Especificidad') +
ylab('Sensibilidad') +
```

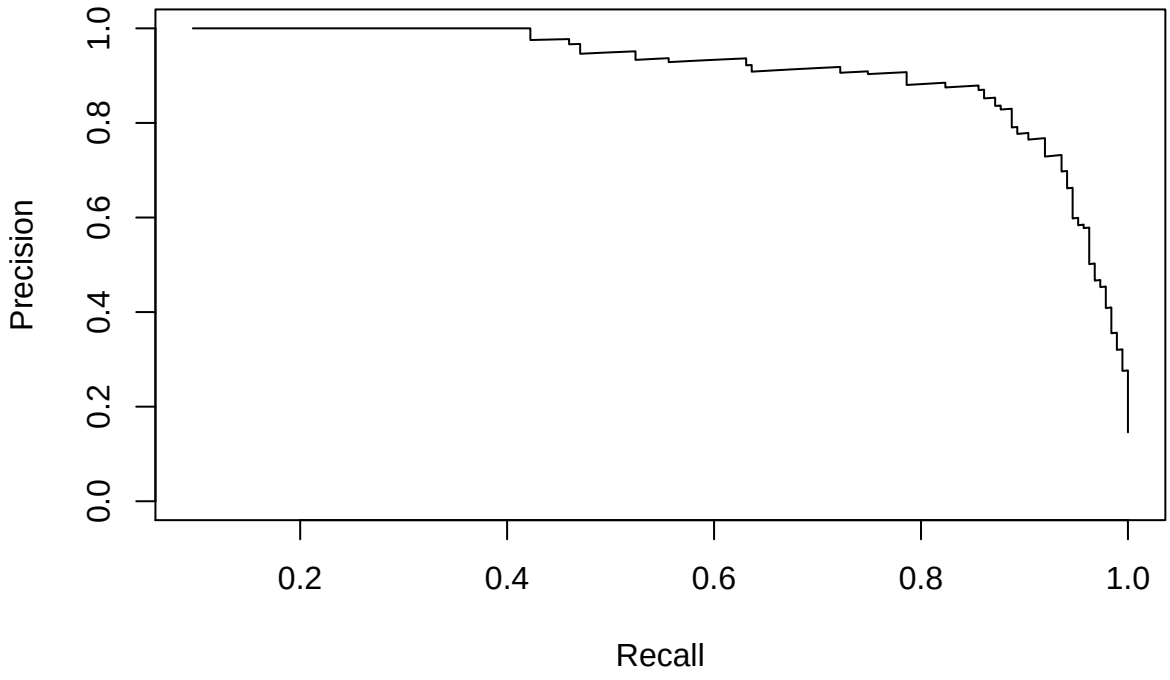
```
scale_x_continuous(breaks = seq(0,1,0.25), labels = seq(0,1,0.25) * 100) +
scale_y_continuous(breaks = seq(0,1,0.25), labels = seq(0,1,0.25) * 100) +
  annotate("text", x = .25, y = .75,size = 5,colour = 'red', label = paste("AUC =", round(roc1$auc,4)))
```

```
Scale for 'x' is already present. Adding another scale for 'x', which will
replace the existing scale.
```



A continuación, calculamos la curva de Precision - Recall, que por tratarse de un dataset desbalanceado, mide más estrictamente la calidad del modelo.

```
predobj <- prediction(data_h$prob, data_h$consolidation)
perf <- performance(predobj,"prec", "rec")
plot(perf,ylim=c(0,1))
```



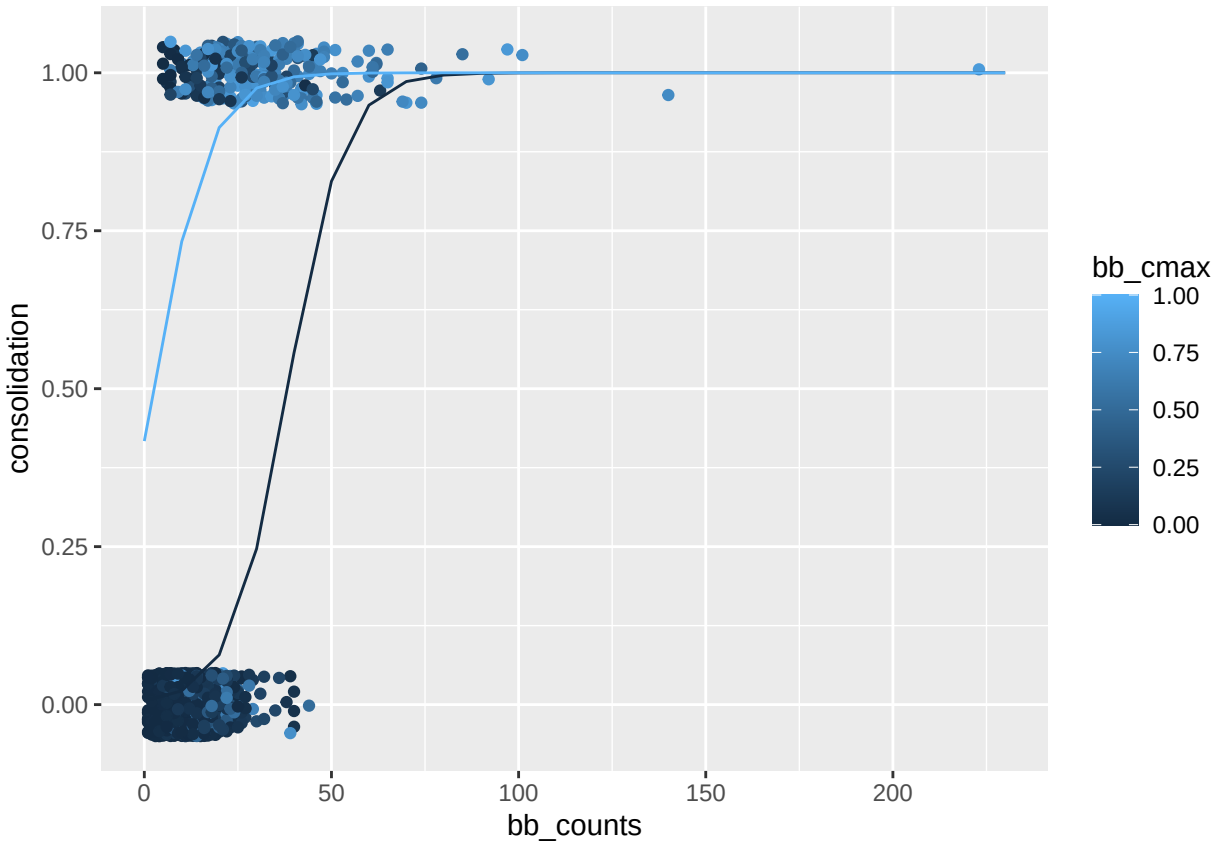

```
x = perf@x.values[[1]]
y = perf@y.values[[1]]

idx = 2:length(x)
testdf=data.frame(recall = (x[idx] - x[idx-1]), precision = (y[idx] + y[idx-1]))

# Ignore NAs
testdf = subset(testdf, !is.na(testdf$precision))
(AUPRC = sum(testdf$recall * testdf$precision)/2)
```

```
[1] 0.8213563
```

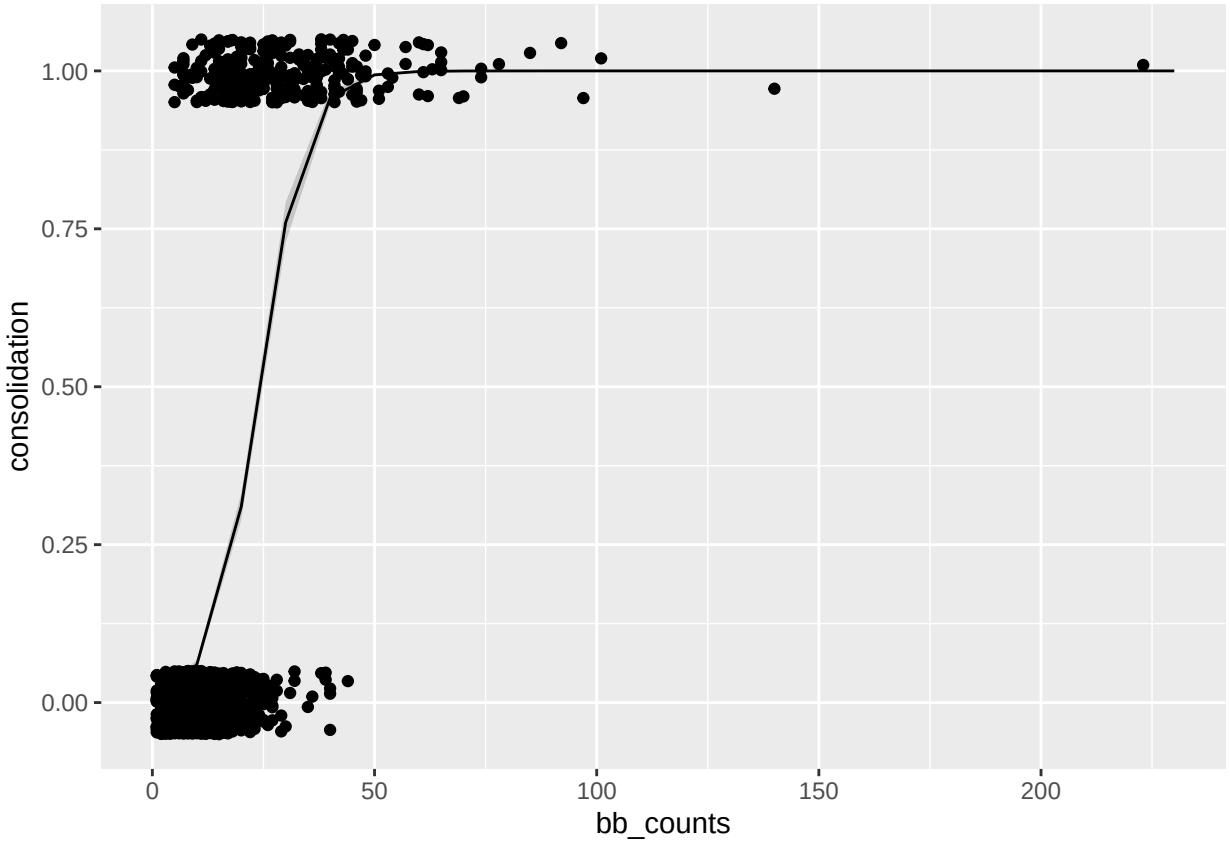
```
# ROC Curve
(AUROC <- performance(predobj,"auc")@y.values)
```

```
[[1]]
[1] 0.9796674
```

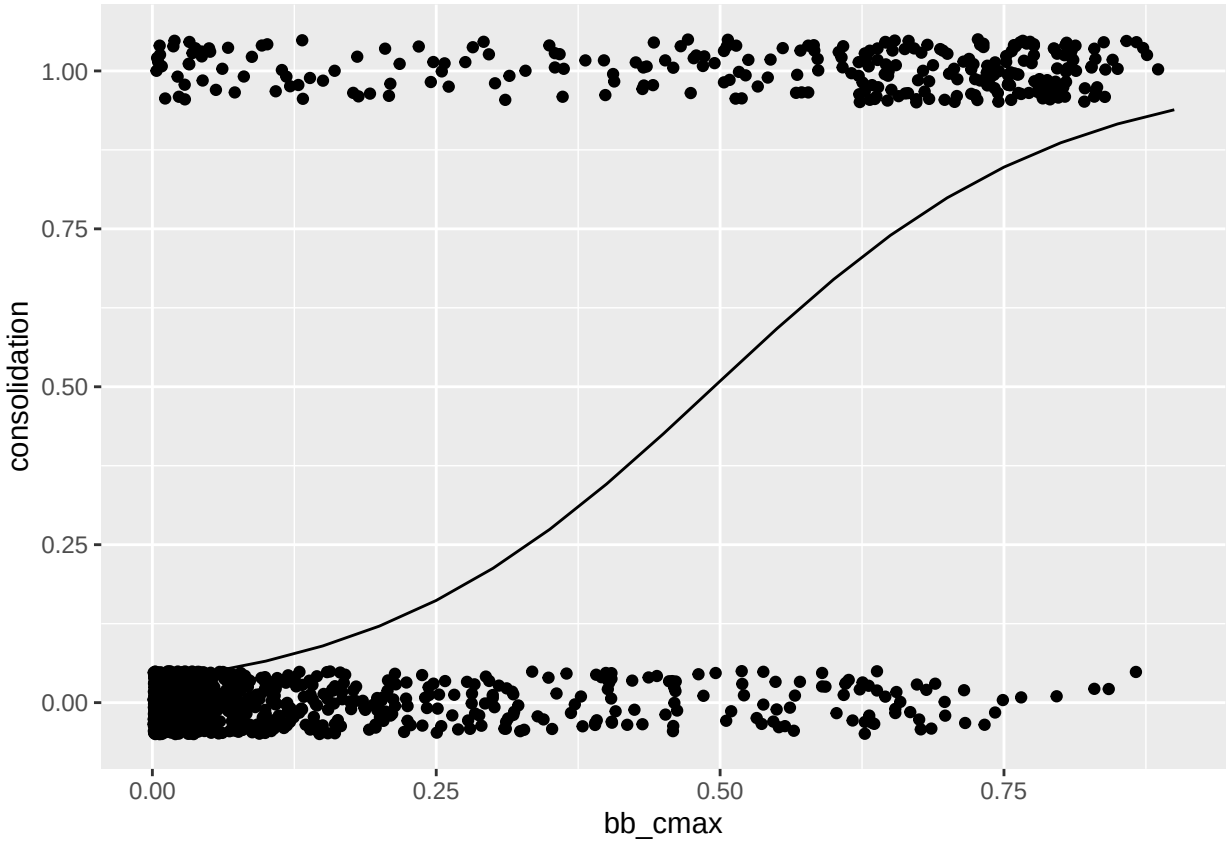
```
ggPredict(m1,colorn=2)
```




```
ggPredict(m5,se=TRUE)
```




```
ggPredict(m6)
```



Podemos ver que en ambos casos no existe separatividad lineal, es decir, sin importar el umbral que se escoja el modelo no es capaz de aislar perfectamente el grupo de Consolidaciones del de No findings.