Spam: modelo logístico y curvas ROC

Leemos los datos

```
library(readr)
library(tidyr)
library(dplyr)
spam_entrena <- read_csv('./datos/spam-entrena.csv')
spam_prueba <- read_csv('./datos/spam-prueba.csv')
```

1. Modelo sólo utilizando las variables caracteres

Vamos a utilizar un modelo logístico para estimar si es spam o no en función de las variables cfsc, cfpar, etc

```
logistico <- glm(spam ~ cfsc+cfpar+cfbrack+cfexc+cfdollar+
                   cfpound,data=spam_entrena, family = 'binomial')
summary(logistico)
```

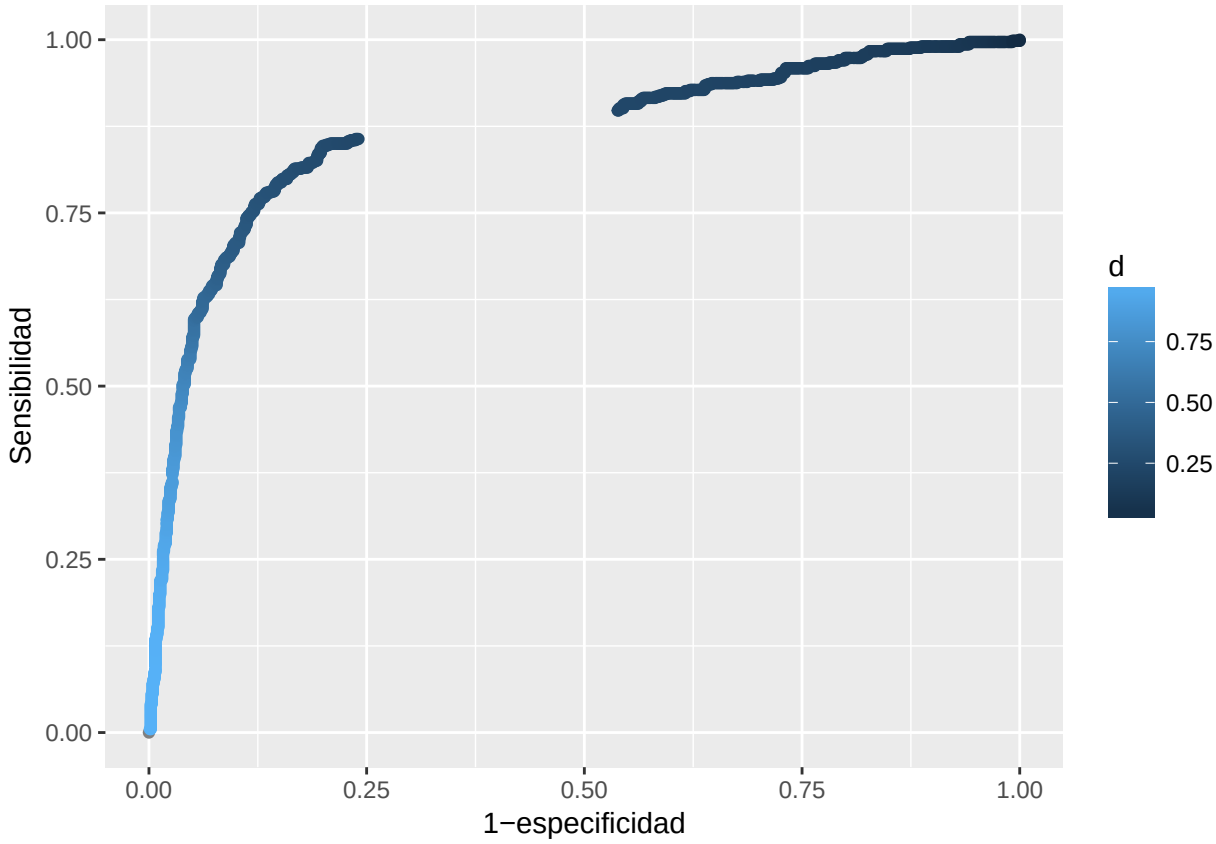
```
##
## Call:
## glm(formula = spam ~ cfsc + cfpar + cfbrack + cfexc + cfdollar +
##     cfpound, family = "binomial", data = spam_entrena)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.4904  -0.7270  -0.5603   0.6012   2.5107
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.19573    0.06619 -18.066  < 2e-16 ***
## cfsc        -1.11752    0.52647  -2.123   0.0338 *
## cfpar       -1.65865    0.30171  -5.497 3.85e-08 ***
## cfbrack     -2.02263    1.03569  -1.953   0.0508 .
## cfexc        1.63224    0.13929  11.719  < 2e-16 ***
## cfdollar    14.87256    0.84558  17.589  < 2e-16 ***
## cfpound      0.65780    0.27429   2.398   0.0165 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4110.8  on 3066  degrees of freedom
## Residual deviance: 2829.9  on 3060  degrees of freedom
## AIC: 2843.9
##
## Number of Fisher Scoring iterations: 7
```

```
preds_prueba <- predict(logistico,newdata = spam_prueba , type="response")
preds_entrena<-predict(logistico, newdata=spam_entrena,type="response")
```

Construimos la curva ROC (prueba):

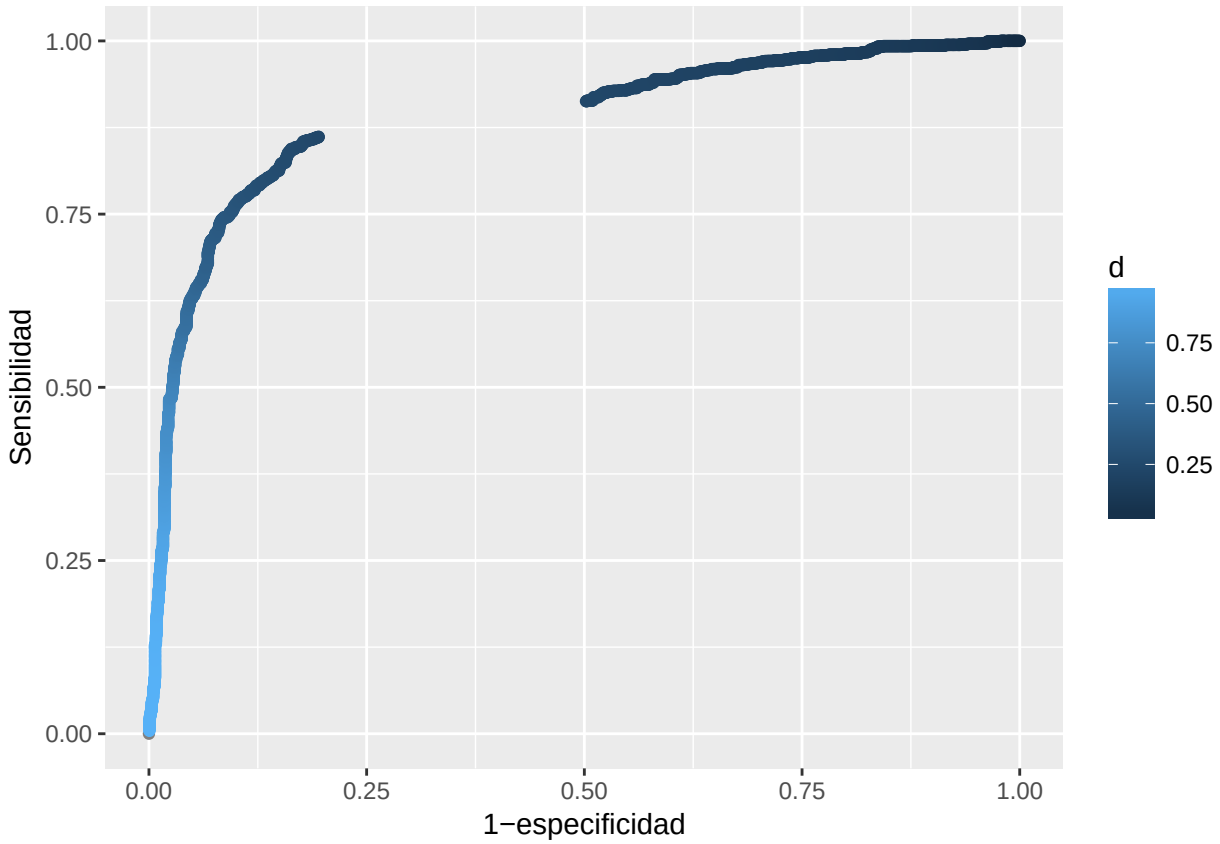
```
library(ROCR)
library(ggplot2)
```

```
pred_rocr_1 <- prediction(preds_prueba, spam_prueba$spam)
perf_1 <- performance(pred_rocr_1, measure = "sens", x.measure = "fpr")
graf_roc_1 <- data_frame(tfp = perf_1@x.values[[1]], sens = perf_1@y.values[[1]],
                         d = perf_1@alpha.values[[1]])

ggplot(graf_roc_1, aes(x = tfp, y = sens, colour=d)) + geom_point() +
  xlab('1-especificidad') + ylab('Sensibilidad')
```



Curva ROC (entrenamiento):

```
library(ROCR)
pred_rocr_2 <- prediction(preds_entrena, spam_entrena$spam)
perf_2 <- performance(pred_rocr_2, measure = "sens", x.measure = "fpr")
graf_roc_2 <- data_frame(tfp = perf_2@x.values[[1]], sens = perf_2@y.values[[1]],
                         d = perf_2@alpha.values[[1]])

ggplot(graf_roc_2, aes(x = tfp, y = sens, colour=d)) + geom_point() +
  xlab('1-especificidad') + ylab('Sensibilidad')
```



2. Modelo utilizando todas las variables

Constuimos el modelo usando todas las variables:

```
logistico_todas <- glm(spam ~.,data=spam_entrena, family = 'binomial')
summary(logistico_todas)
```

```
##
## Call:
## glm(formula = spam ~ ., family = "binomial", data = spam_entrena)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.0931  -0.2097   0.0000   0.1202   4.3396
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.553e+00  2.123e-01  -7.314 2.59e-13 ***
## X1           -1.956e-05  8.214e-05  -0.238 0.811744
## wfmake       -4.702e-01  2.880e-01  -1.633 0.102559
## wfaddress    -1.579e-01  1.006e-01  -1.570 0.116342
## wfall         3.001e-01  1.352e-01   2.219 0.026455 *
## wf3d          2.400e+00  1.780e+00   1.348 0.177527
## wfour         5.472e-01  1.188e-01   4.607 4.09e-06 ***
## wfover        6.504e-01  2.657e-01   2.448 0.014365 *
## wfremove      2.357e+00  3.984e-01   5.916 3.30e-09 ***
## wfinternet    4.087e-01  1.893e-01   2.159 0.030848 *
```

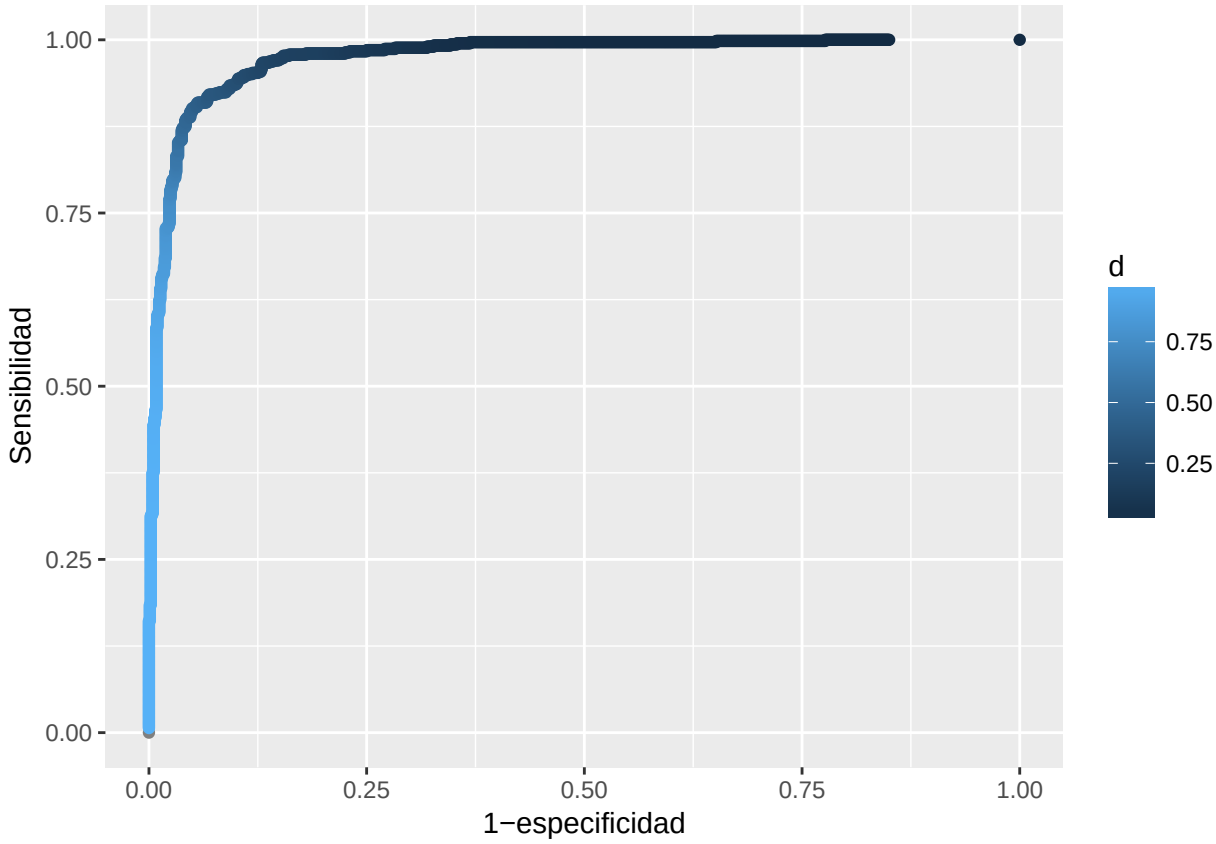
```
## wforder       7.279e-01  3.517e-01   2.069 0.038508 *
## wfmail        7.988e-02  8.375e-02   0.954 0.340225
## wfreceive     5.708e-02  3.369e-01   0.169 0.865441
## wfwill       -1.366e-01  9.107e-02  -1.500 0.133721
## wfpeople     -7.342e-03  2.702e-01  -0.027 0.978327
## wfreport      1.046e-01  1.447e-01   0.723 0.469660
## wfaddresses   1.056e+00  7.425e-01   1.423 0.154778
## wffree        1.041e+00  1.680e-01   6.193 5.90e-10 ***
## wfbusiness    9.442e-01  2.592e-01   3.643 0.000270 ***
## wfemail       1.535e-01  1.395e-01   1.101 0.270917
## wfyou         9.202e-02  4.298e-02   2.141 0.032259 *
## wfcredit      8.242e-01  5.653e-01   1.458 0.144863
## wfyour        2.059e-01  6.332e-02   3.251 0.001149 **
## wffont        2.135e-01  2.381e-01   0.897 0.369970
## wf000         2.593e+00  6.099e-01   4.251 2.13e-05 ***
## wfmoney       3.045e-01  1.432e-01   2.126 0.033543 *
## wfhp         -1.603e+00  3.262e-01  -4.915 8.87e-07 ***
## wfhpl        -1.153e+00  5.198e-01  -2.219 0.026478 *
## wfgeorge     -7.737e+00  2.021e+00  -3.829 0.000129 ***
## wf650         3.960e-01  2.080e-01   1.904 0.056869 .
## wflab        -2.863e+00  2.417e+00  -1.185 0.236140
## wflabs       -6.206e-01  4.239e-01  -1.464 0.143188
## wftelnet     -1.424e-01  4.240e-01  -0.336 0.736899
## wf857         1.900e+00  3.949e+00   0.481 0.630338
## wfdata       -6.033e-01  3.352e-01  -1.800 0.071911 .
## wf415        -7.082e+00  1.290e+01  -0.549 0.582966
## wf85         -1.784e+00  9.885e-01  -1.805 0.071140 .
## wftechnology  7.472e-01  3.574e-01   2.091 0.036546 *
## wf1999        3.531e-01  2.595e-01   1.361 0.173564
## wfparts       7.806e-01  1.326e+00   0.589 0.555906
## wfpm         -1.108e+00  5.313e-01  -2.085 0.037068 *
## wfdirect     -1.847e-01  4.613e-01  -0.400 0.688867
## wfcs         -2.592e+02  9.293e+03  -0.028 0.977747
## wfmeeting    -2.834e+00  1.023e+00  -2.770 0.005609 **
## wforiginal   -1.561e+00  1.029e+00  -1.517 0.129190
## wfproject    -1.905e+00  7.439e-01  -2.561 0.010436 *
## wfre         -7.577e-01  2.011e-01  -3.768 0.000164 ***
## wfedu        -1.199e+00  3.148e-01  -3.809 0.000140 ***
## wftable      -2.696e+00  2.282e+00  -1.181 0.237487
## wfconference -3.969e+00  2.158e+00  -1.839 0.065881 .
## cfsc         -1.448e+00  6.496e-01  -2.229 0.025846 *
## cfpar        -7.899e-01  5.000e-01  -1.580 0.114184
## cfbrack      -6.663e-01  9.273e-01  -0.719 0.472409
## cfexc         3.149e-01  9.971e-02   3.159 0.001585 **
## cfdollar      6.172e+00  8.771e-01   7.037 1.97e-12 ***
## cfpound       2.951e+00  7.802e-01   3.782 0.000155 ***
## crlaverage   -8.439e-03  1.885e-02  -0.448 0.654441
## crllongest    8.201e-03  2.820e-03   2.908 0.003635 **
## crltotal      9.677e-04  2.791e-04   3.468 0.000525 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 4110.8  on 3066  degrees of freedom
## Residual deviance: 1235.9  on 3008  degrees of freedom
## AIC: 1353.9
##
## Number of Fisher Scoring iterations: 22
```

```
preds_prueba_todas <- predict(logistico_todas,newdata = spam_prueba , type="response")
preds_entrena_todas<-predict(logistico_todas, newdata=spam_entrena,type="response")
```

Construimos la curva ROC (prueba) utilizando todas las variables:

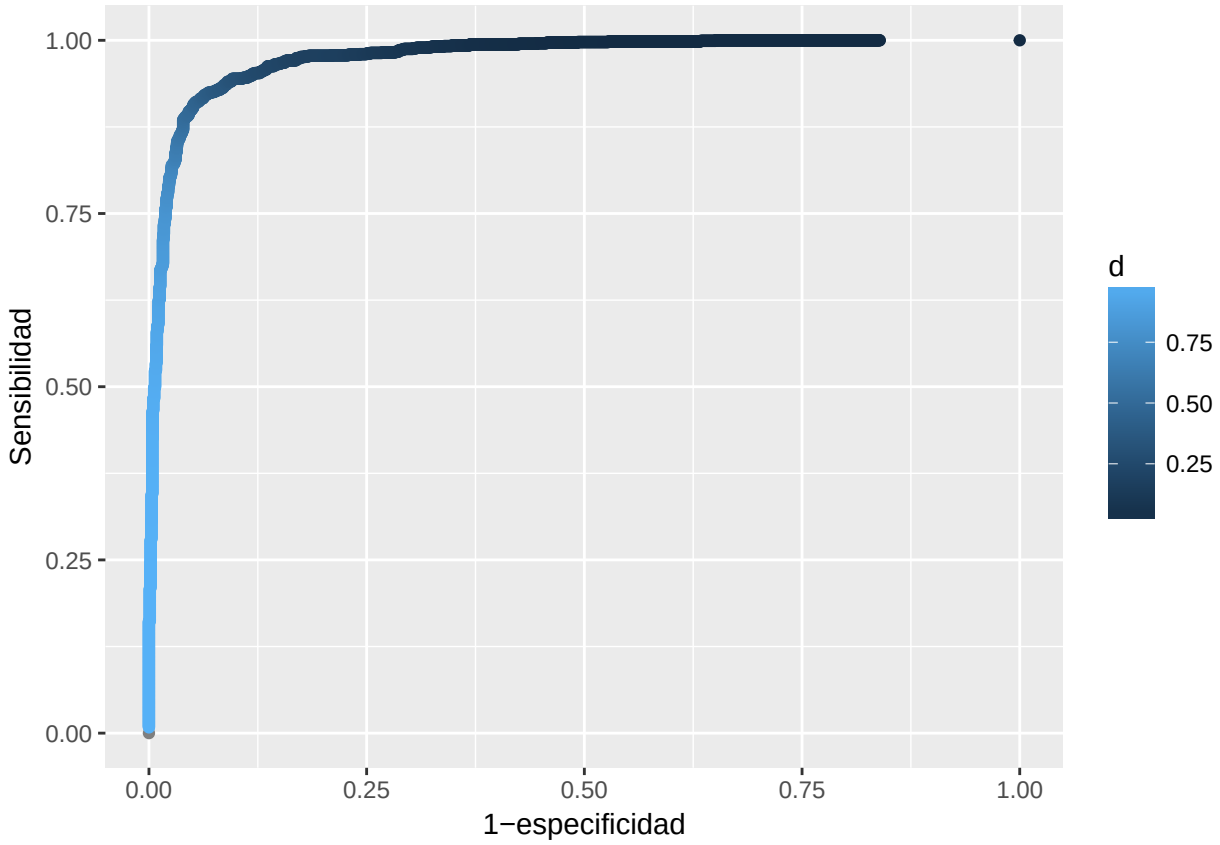
```
library(ROCR)
library(ggplot2)
pred_rocr_3 <- prediction(preds_prueba_todas, spam_prueba$spam)
perf_3 <- performance(pred_rocr_3, measure = "sens", x.measure = "fpr")
graf_roc_3 <- data_frame(tfp = perf_3@x.values[[1]], sens = perf_3@y.values[[1]],
                         d = perf_3@alpha.values[[1]])

ggplot(graf_roc_3, aes(x = tfp, y = sens, colour=d)) + geom_point() +
  xlab('1-especificidad') + ylab('Sensibilidad')
```



Curva ROC (entrenamiento) utilizando todas las variables:

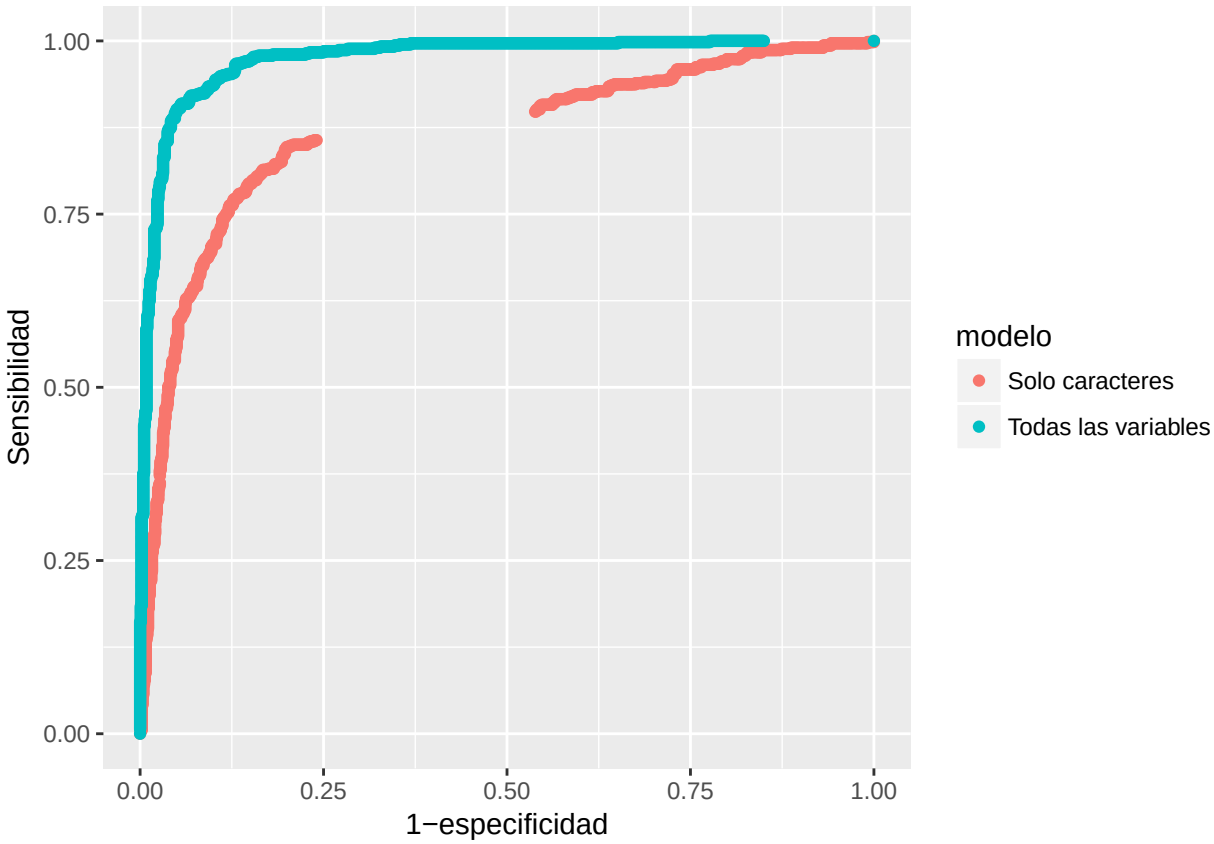
```
library(ROCR)
pred_rocr_4 <- prediction(preds_entrena_todas, spam_entrena$spam)
perf_4 <- performance(pred_rocr_4, measure = "sens", x.measure = "fpr")
graf_roc_4 <- data_frame(tfp = perf_4@x.values[[1]], sens = perf_4@y.values[[1]],
                         d = perf_4@alpha.values[[1]])
```

```r
ggplot(graf_roc_4, aes(x = tfp, y = sens, colour=d)) + geom_point() +
  xlab('1-especificidad') + ylab('Sensibilidad')
```



3. Graficamos las dos curvas ROC de prueba:

```r
graf_roc_3$modelo <- 'Todas las variables'
graf_roc_1$modelo <- 'Solo caracteres'
graf_roc <- bind_rows(graf_roc_1, graf_roc_3)

ggplot(graf_roc, aes(x = tfp, y = sens, colour = modelo)) + geom_point() +
  xlab('1-especificidad') + ylab('Sensibilidad')
```



Resulta superior el modelo utilizando todas las variables, en pimera instancia pues el que sólo tiene caracteres no completa la curva ROC y en segundo lugar el clasificador que usa todas las variables domina siempre al clasificador que sólo utiliza las variables caracteres; es decir, para cualquier punto de corte siempre existe un clasificador en la curva azul (todas las variables) que domina al que sólo tiene la variable caracter.

4. Punto de corte apropiado para hacer un filtro de spam

En mi opinion resulta más grave que un mail sea catalogado como spam cuando no lo es pues en este caso podría ser un mail importante que se fue directo a la papelera de reciclaje sin que lo hayamos siquiera visto; es decir, me parece que son mucho más graves los falsos positivos. En virtud de lo anterior, considero que debemos escoger un punto de corte con especifidad más grande y con sensibilidad más chica por lo que escogeré como punto de corte $d = 0.8$

La tabla considerando este punto de corte quedaría como sigue:

```
table(preds_prueba_todas > 0.8, spam_prueba$spam)
```

```
##
##           0   1
##   FALSE 908 165
##   TRUE   19 442
```

Y la proporción con $d = 0.8$:

```
prop.table(table(preds_prueba_todas > 0.8, spam_prueba$spam),2)
```

```
##
##                   0          1
##   FALSE 0.97950378 0.27182867
```

```
##   TRUE  0.02049622 0.72817133
```

A continuación incluyo la table con $d = 0.5$ para contrastar:

```
table(preds_prueba_todas>.5, spam_prueba$spam)
```

```
##
##           0   1
##   FALSE 891  77
##   TRUE   36 530
```

Finalmente la proporción con $d = 0.5$:

```
prop.table(table(preds_prueba_todas > 0.5, spam_prueba$spam),2)
```

```
##
##                   0          1
##   FALSE 0.96116505 0.12685338
##   TRUE  0.03883495 0.87314662
```

Como conclusión podemos establecer que aunque eliminamos casi por completo los falsos positivos, vamos a estar recibiendo mucho spam, por lo que el costo de no clasificar un correo "bueno" como spam va a ser tener que estar trasladando manualmete el spam a la papelera de reciclaje.