MODELOS DATA FRAME TITANIC

Federico Fernández Moreno
Jorge Ramírez Carrasco

Summer Camp 2014

Creación de Modelos

En primer lugar probaremos a obtener modelos con distintas técnicas de modelado.

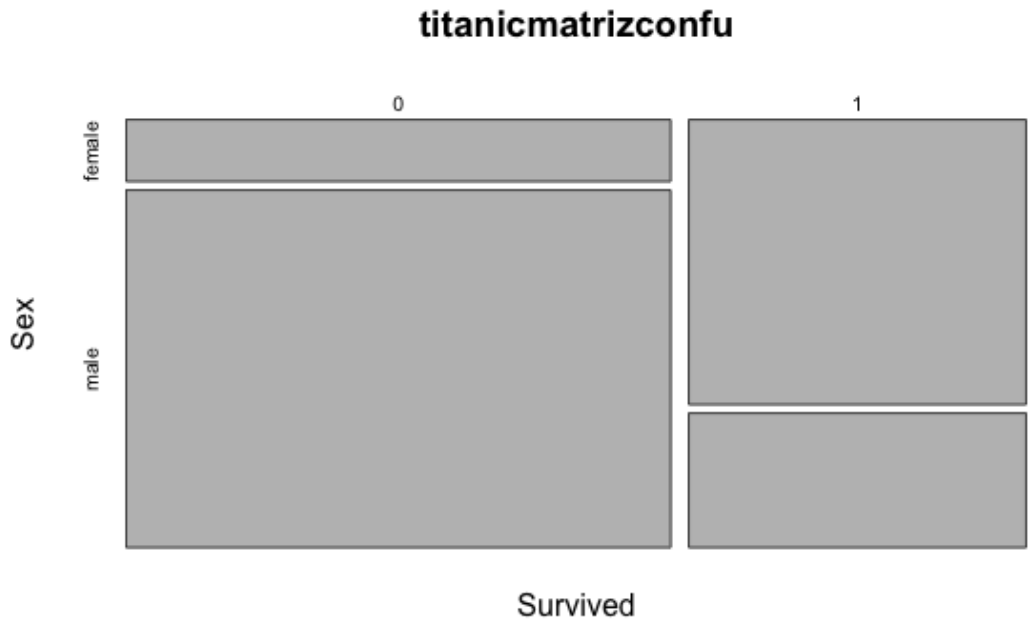
1. CLASIFICACIÓN

En la clasificación podemos obtener valores que pertenezcan a un conjunto ("factor") a partir del modelo. En nuestro caso trataremos de predecir la columna que indica la supervivencia

- Modelos de variable única:

Hemos realizado distintas matrices de confusión para visualizar relaciones entre variables que nos ayuden a establecer relaciones.

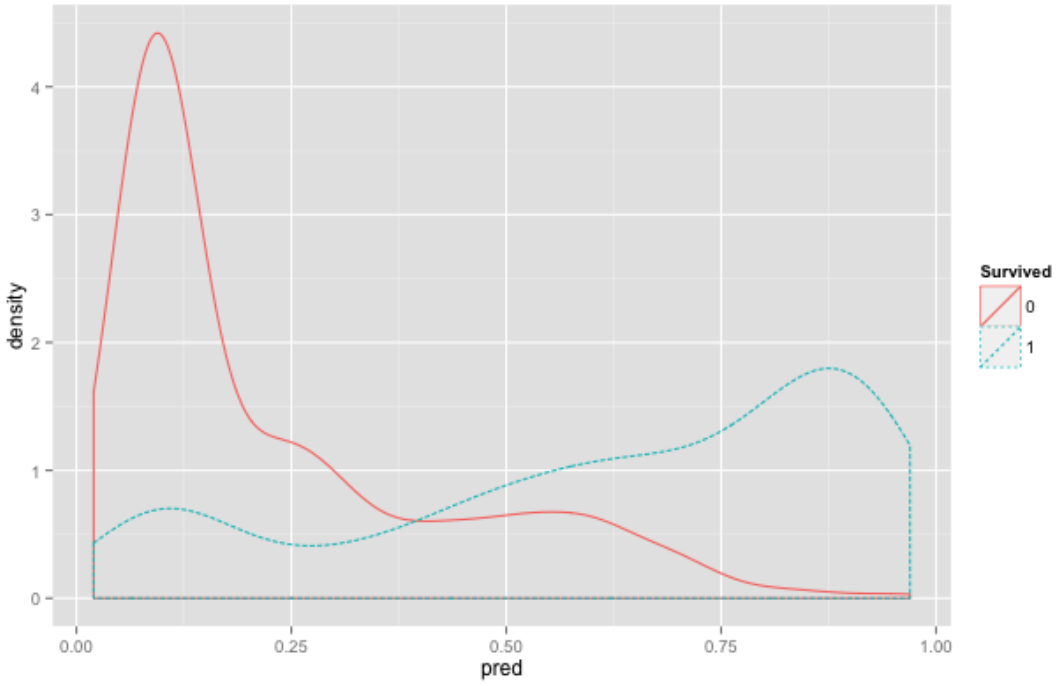
```
#..........Confussion Matrix:

#Survived vs. Sex#
titanicmatrizconfu                                    <-
table(Survived=titanic$Survived,Sex=titanic$Sex)
```



- Regresión Logística:

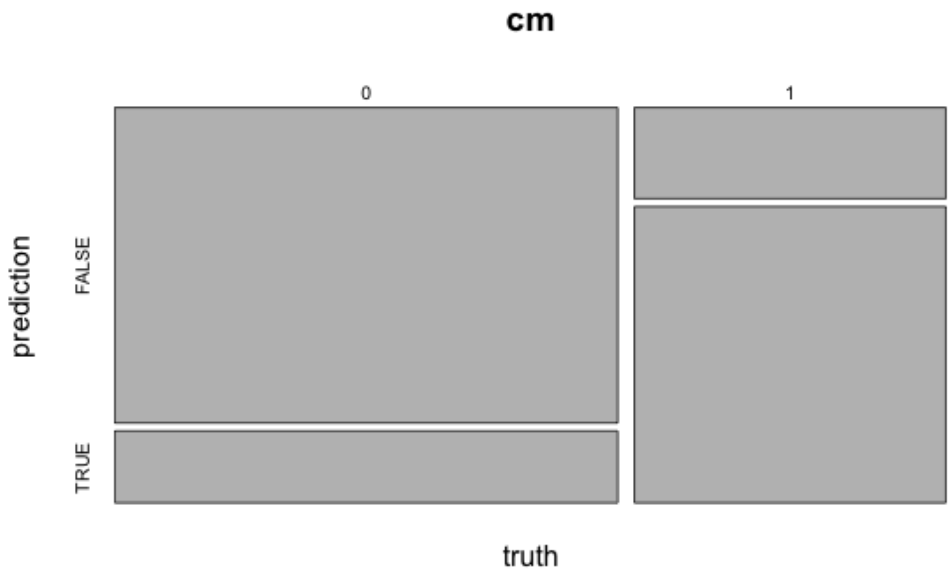
Hemos obtenido distintos modelos de regresión logística, probando con distintas variables y umbrales como explicaremos en la parte de valoración.

```
#......Logistic Regression Model: Survived = f(Pclass, Sex, Age)

formula <- paste("Survived",paste(c("Pclass","Sex","Age"),collapse='+'),sep=' ~ ')

logmodel <- glm(formula,data=titanic,family=binomial(link='logit'))

titanic$pred<- predict(logmodel,newdata=titanic,type='response')

library(ggplot2)

ggplot(titanic,aes(x=pred,color=Survived,linetype=Survived))+geom_density()
```



Elegimos el umbral en 0.44 con una función para ello y realizamos la predicción

```
cm <- table(truth=titanic$Survived,prediction=titanic$pred>0.44)

print(cm)

plot(cm)
```



- Árboles de decisión:

Con la función rpart() obtenemos árboles de decisión con las variables que le indiquemos.

```
#.........Decision tree Model

library(rpart)
treez<-rpart(Survived ~ Age + Sex,data=titanic)
plot(treez)
text(treez,use.n=TRUE)
```

- Random Forest:

Los árboles de decisión más eficientes son los random forest con los que podemos obtener un modelo.

Existen dos paquetes básicos para crear Random Forests en R:

```
## ...Random Forests método 1

library(randomForest)
set.seed(5123512)

fmodel <- randomForest(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked, data=titanic, importance=TRUE, ntree=8000)

## .... Random Forests método 2
library(party)
set.seed(415)
fit_rand_forest <- cforest(Survived ~ Pclass + Sex + Age + SibSp + Fare+ Parch + Embarked,
               data = titanic, controls = cforest_unbiased(ntree=5000, mtry=3))
```

El segundo método es más preciso que el primero, aunque también consume más memoria (360 Mb frente a 60 Mb). Además, parece que el modelo mejora cuantos más árboles tiene (aumentar *ntree*); por supuesto, esto repercute en la memoria que ocupa la variable correspondiente al modelo.

- Reglas de asociación:

Por último, con respecto a la clasificación hemos usado las llamadas reglas de asociación con las que hemos obtenido reglas clave en función de la supervivencia.

```
#.............Reglas de asociación
```

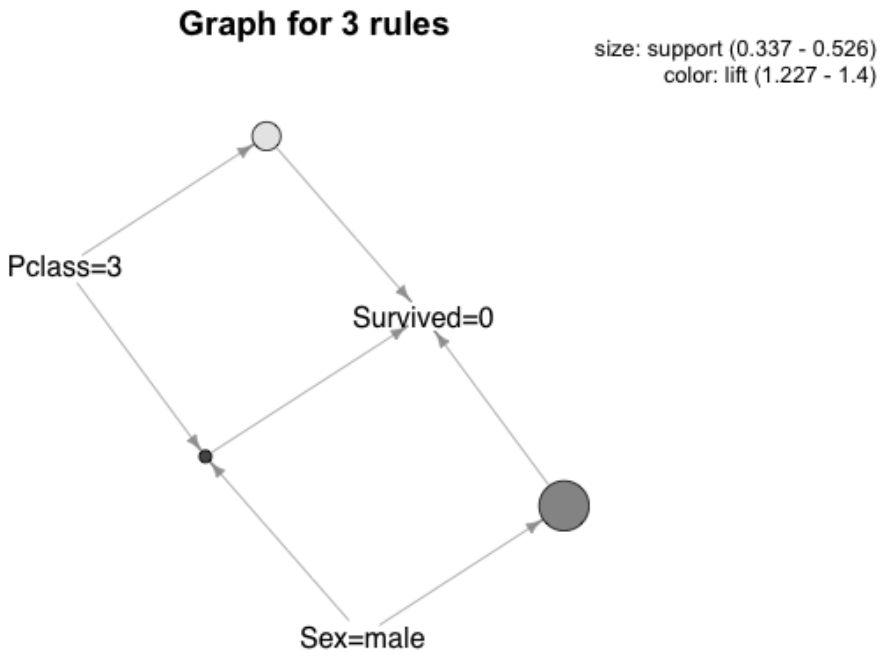
```
library(Matrix)
library(arules)
library(arulesViz)

titanicAsoc <-
titanic[,c("Survived","Sex","Pclass")]

rules <-
apriori(titanicAsoc,parameter=list(support=0.3,con
fidence=0.75),appearance=list(rhs=c("Survived=0","
Survived=1"),default="lhs"))
```

Si subimos el soporte exigimos mayor número de datos a comparar mientras si subimos la confianza exigimos un mayor número de coincidencias.

```
inspect(rules)
```

```
> inspect(rules)
  lhs           rhs            support confidence     lift
1 {Pclass=3} => {Survived=0} 0.4184477  0.7576375 1.226848
2 {Sex=male} => {Survived=0} 0.5264342  0.8110919 1.313407
3 {Sex=male,
   Pclass=3} => {Survived=0} 0.3374578  0.8645533 1.399978
```

```
plot(rules)
```



2. PREDICCIÓN

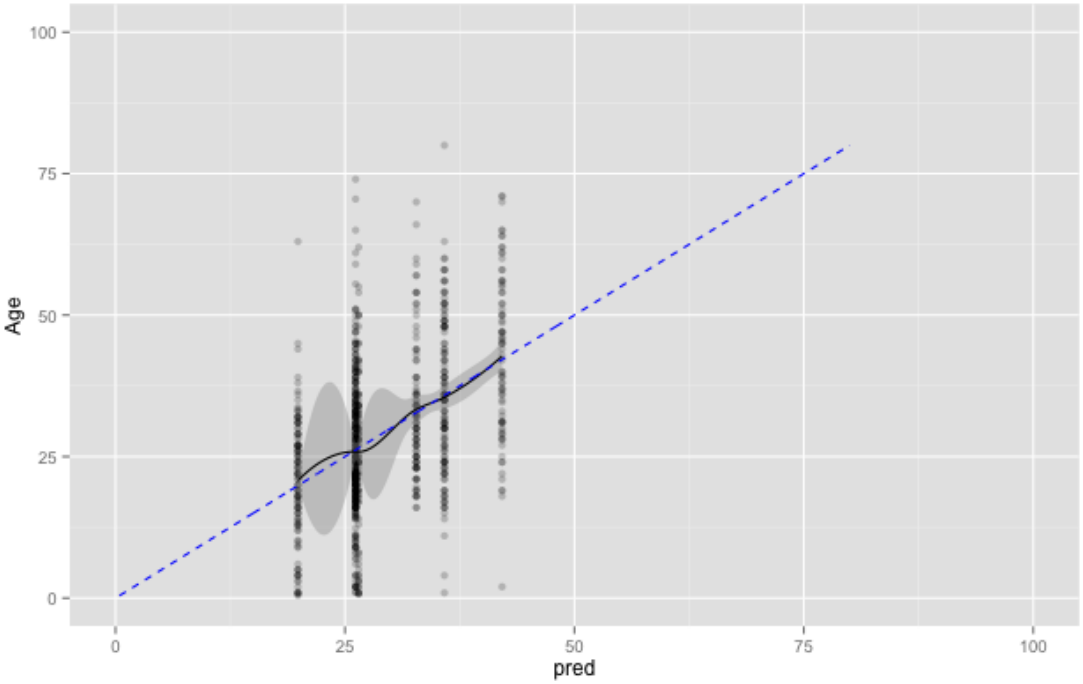
En la predicción podemos obtener valores numéricos a partir del modelo. Para el caso concreto del titanic la variable numérica más significativa es la edad que será la que tomaremos como variable a predecir por los modelos.

- Regresión Lineal:

Hemos realizado en primer lugar un modelo de regresión lineal en el que podemos predecir la edad.

```
#........Linear Regression Model: Age vs Survived & Pclass

linmodel <- lm(Age ~ Survived + Pclass, data=titanic)

titanicpred<-titanic

titanicpred$pred<-

predict(linmodel,newdata=titanic)

library(ggplot2)

ggplot(data=titanicpred,aes(x=pred,y=Age))+
  geom_point(alpha=0.2,color="black")+
  geom_smooth(aes(x=pred,y=Age),color="black")+
```

```
geom_line(aes(x=Age,y=Age),color="blue",linetype=2
)+

scale_x_continuous(limits=c(0,100))+scale_y_contin
uous(limits=c(0,100))
```



- Serie temporal:

En nuestro caso la variable numérica de la edad, no nos dice nada con respecto el tiempo.

```
#............Serie Temporal

library(forecast)
sertempmodel <- auto.arima(titanic$Age)
plot(forecast(sertempmodel,h=1))
```

3. CONCLUSIÓN

En esta primera fase hemos obtenidos diferentes modelos de los cuales los que más se ajustan a nuestras necesidades son regresión logística, principalmente, y random forest. En la siguiente fase valoraremos estos modelos con distintos medidores y ajustaremos hasta tener el mejor modelo posible con el que realizar una predicción.

Valoración de los Modelos

Tras elaborar los modelos anteriores, se ha decidido optar por los modelos basados en regresión logística. En concreto, se han elaborado cuatro modelos, para observar qué ocurre cuando se incluyen distintas variables:

- El modelo 0 es el indicado anteriormente en el apartado *Regresión Logística* de *Creación de los Modelos*, e incluye las variables *Pclass*, *Sex* y *Age*.
- El modelo 1 incluye también las variables *Parch* (padres o hijos en el barco), *Sibsp* (hermanos o esposa/marido en el barco) y *Embarked* (en qué puerto se embarcó).
- El modelo 2 solo añade las variables *Parch* y *Sibsp* respecto al modelo 0.
- El modelo 3 solo añade la variable *Parch* respecto al modelo 0.

Cálculo de métricas y elección de modelo

Para comparar los modelos, y elegir el más adecuado, se ha obtenido la matriz de confusión que resulta de hacer una predicción con los datos de entrenamiento, y se han calculado las métricas precisión, recall, enrichment, specificity, accuracy, false-

		Prediction	
		0	1
Actual	0	TN	FP
	1	FN	TP

positive rate y false-negative rate.

Se ha obtenido la siguiente tabla:

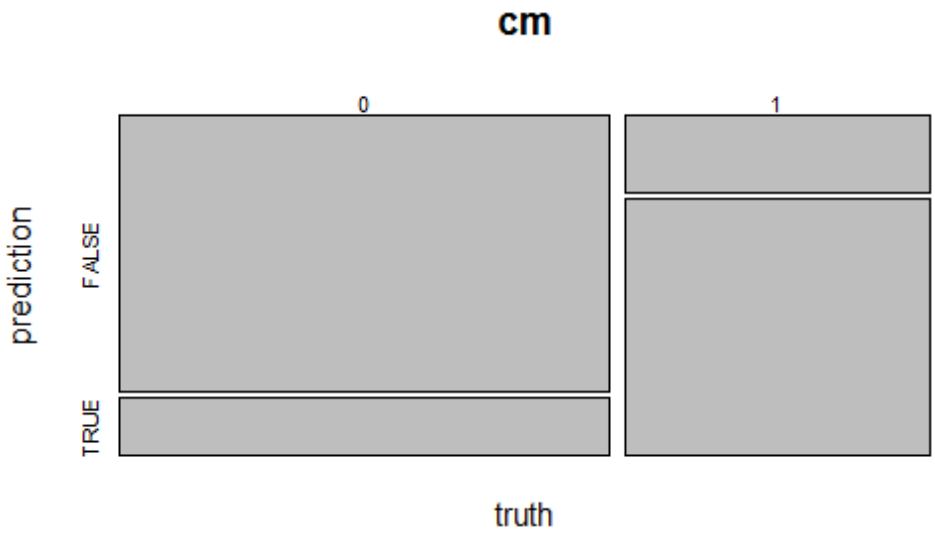
	prec	rec	enrich	spec	accuracy	fpr	fnr
#0	0.7285319	0.7735294	0.5269852	0.8214936	0.8031496	0.1785064	0.2264706
#1	0.7285714	0.7500000	0.5270138	0.8269581	0.7975253	0.1730419	0.2500000
#2	0.7298851	0.7470588	0.5279640	0.8287796	0.7975253	0.1712204	0.2529412
#3	0.7440476	0.7352941	0.5382086	0.8433515	0.8020247	0.1566485	0.2647059

A la vista de estos parámetros, parece que el método que más se ajusta al deseado es el método #0, ya que su *precisión* y su *recall* están equilibrados, y la *accuracy* es la máxima de todos los modelos, lo cual indica que, en el 80% de las veces que digamos que alguien va a sobrevivir, esto será cierto. Parece que, dada la especificación del modelo deseado, tiene más sentido permitir un cierto error al predecir una muerte (si luego no se produce, es un error que no repercute en el "cliente"), mientras que un error al predecir una supervivencia es, obviamente, determinante para el "cliente".

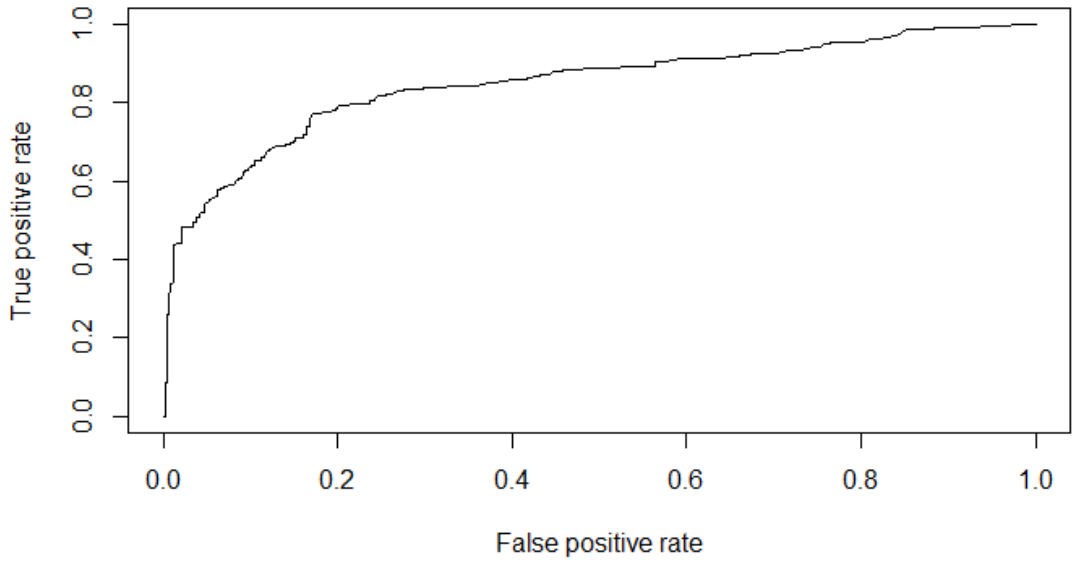
Caracterización del modelo elegido

Una vez elegido el método de regresión logística que se va a utilizar, van a calcularse otros parámetros que permitan su completa caracterización.

- Matriz de confusión: el modelo tiene la matriz de confusión representada en la siguiente figura



- Curva ROC: la curva de la "Característica Operativa del Receptor" (en inglés, Receiver Operating Characteristic) queda como sigue.



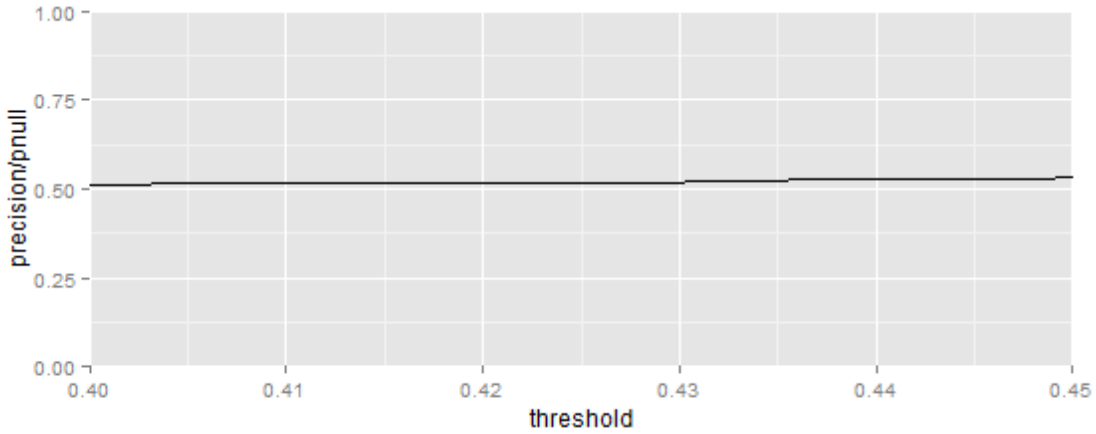
La curva óptima es la que más se parece a un escalón (con área bajo la curva, AUC = 1), y la curva que correspondería a un modelo completamente al azar es la recta y=x (AUC = 0,5). Este modelo tiene un AUC = 0,8488.

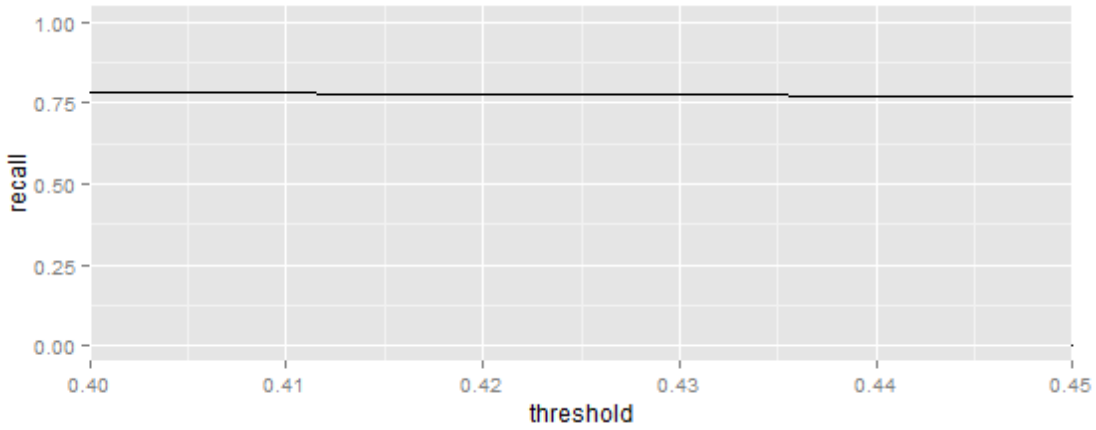
- Log-Likelihood: la verosimilitud logarítmica (en inglés, log-likelihood) es -399.83. A

primera vista parece un valor muy alejado del óptimo (valores próximos a cero son los deseables), pero si se realiza el cálculo en términos de los datos que se poseen (verosimilitud logarítmica "relativa") se obtiene el valor -0.45. Además, la verosimilitud logarítmica del "mejor modelo nulo" (aquel que solo tiene en cuenta una variable y que predice a partir de la media de esa variable) es -591.41, con lo que el modelo propuesto mejora notablemente este parámetro.

Umbral de decisión óptimo (threshold)

Por último, se ha estudiado cuál es el umbral de decisión óptimo. La función *predict* de R permite calcular probabilidades de estar en una u otra clase (supervivencia o no supervivencia), pero es necesario traducir esta probabilidad a pertenencia o no pertenencia a la clase. Las siguientes gráficas muestran cómo cambia la *precisión respecto al modelo nulo* y la métrica *recall* en función del umbral (*threshold*) que se tome:





Parece que cualquier valor de umbral entre 0.40 y 0.45 no influye en dichas métricas, por lo que se ha elegido el valor que optimiza la matriz de confusión (menor número de FN y FP): 0.44.

Aunque son las mismas que se indicaron en una tabla anterior, los valores de las métricas que se obtienen para este valor de umbral son:

prec	rec	enrich	spec	acc	fpr	fnr
0.7285	0.7735	0.5270	0.8215	0.8031	0.1785	0.2264

Influencia de los coeficientes

Además, se ha comprobado que los coeficientes que utiliza el modelo influyen todos ellos en la predicción, ya que su valor *Pr(>|z|)* es siempre mucho menor que 0.05:

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept)  3.70556    0.37399   9.908  < 2e-16 ***
Pclass2     -1.20304    0.26232  -4.586 4.52e-06 ***
Pclass3     -2.47923    0.25382  -9.768  < 2e-16 ***
Sexmale     -2.57833    0.18747 -13.753  < 2e-16 ***
Age         -0.03682    0.00735  -5.010 5.45e-07 ***
```

Otros parámetros

El criterio de información de Aikake (AIC) para este modelo es 809.6, el menor de todos los modelos de regresión logística planteados, lo que también indica que se ha elegido el modelo óptimo.

El valor de F-Score para este modelo es 5.

Predicción con datos de test

Una vez hemos comparado modelos y elegido el que creemos más apropiado, realizamos un test con un .csv con datos que contienen toda la información salvo la supervivencia la cual tendremos que predecir con nuestro modelo.

En primer lugar realizamos el acondicionamiento de los datos como hicimos en la exploración.
A continuación tomamos el modelo y predecimos.

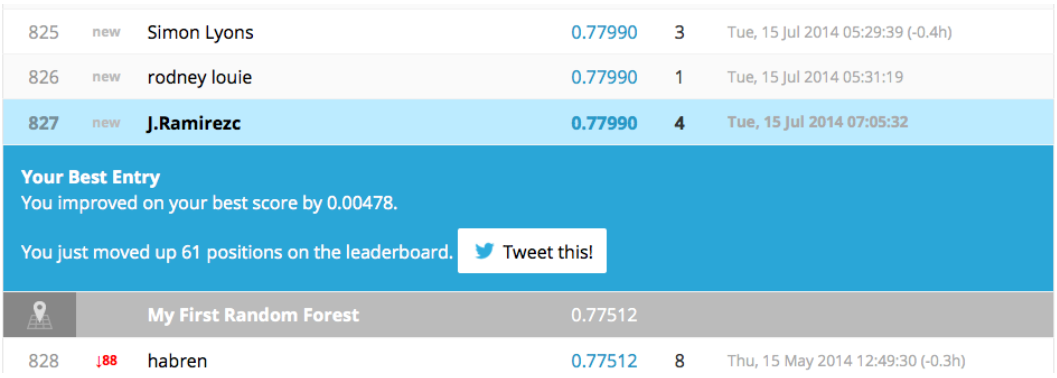
```
#Creamos modelo

#.........Logistic Regression Model: Survived =
f(Pclass, Sex, Age)

##....Predecimos

titanic$Survived<-
ifelse(predict(logmodel,newdata=titanic,type='resp
onse')>0.44,1,0)

#....Creamos dataframe de test

titanic_test<-
titanic[c("PassengerId","Survived")]
write.csv(titanic_test,file="titanic_test.csv",row
.names=FALSE)
```

Por último subimos nuestros resultados a Kaggle en un csv con solo la Id del pasajero y el campo "Survived" para comprobar las predicciones de nuestro modelo con los datos de test almacenados en la plataforma.

1258	↓124	caldaria1	0.73684	2	Fri, 20 Jun 2014 19:04:34 (-0.1h)
1259	↓124	Yuliya	0.73684	2	Sun, 22 Jun 2014 18:15:54 (-0.1h)
1260	↓101	alfa2plus	0.73684	4	Sun, 06 Jul 2014 16:35:01 (-0h)
1261	new	Szabolcs75	0.73684	2	Thu, 10 Jul 2014 20:35:45
1262	new	**J.Ramirezc**	**0.73684**	**2**	**Fri, 11 Jul 2014 10:25:39 (-0.3h)**
1263	new	vsurjaninov	0.73684	6	Fri, 11 Jul 2014 16:07:10 (-0.5h)
1264	new	Sean McFarland	0.73684	1	Sat, 12 Jul 2014 00:10:46

Esta predicción la hemos mejorado utilizando random forests. En primer lugar hemos utilizado para predecir un modelo con la librería random forest la cual mejora nuestros resultados

```
##....Predecimos

titanic_test$Survived<-                predict(fmodel,
titanic_test)
```



A continuación utilizamos la librería party para generar nuestro random forest y las predicciones obtenidas son más acertadas todavía

```
##....Predecimos

titanic_test$Survived <- predict(fit_rand_forest,
titanic_test, OOB=TRUE, type = "response")
```