Minería de Datos - Práctica Calificada N.3

HUERTAS, ANTHONY[1,*]

[1] *Maestría en Estadística, Escuela de Posgrado, Pontificia Universidad Católica del Perú, Lima, Perú*
[*] *Cod: 20173728*

Compiled July 8, 2018

Profesor: Luis Benites

Utilizando el conjunto de datos de titanic_train del paquete titanic, ajuste los modelos de clasificación para predecir si el pasajero sobrevivió o no. Explore todos los modelos visto en clase. Describa sus hallazgos.

EXTRACCIÓN Y LIMPIEZA DEL CONJUNTO DE DATOS.

En esta sección, se importarán los datos de entrenamiento y evaluación, sin embargo se observará la presencia de valores "NA", a lo que se imputarían valores adecuados, como de otros factores necesarios a eliminar, esto con el objetivo de tener datos completos necesarios para poder iniciar el análisis respectivo tanto de las variables en estudio como de los modelos de clasificación, que seran vistas luego.

Las librerías que serán usadas son las siguientes

Listing 1. Librerías

```
pacman::p_load(titanic,visdat,ggplot2,dplyr,e1071,ROCR,kernlab,randomForest,mice,vcd,MASS,aod)
```

Se importan datos de entrenamiento titanic_train y evaluación titanic_test provenientes de la librería titanic, con adición de una representación gráfica del porcentaje de valores faltantes en cada subconjunto de datos.

Listing 2. Titanic: Datos de entrenamiento.

```
head(titanic_train,3)
vis_miss(titanic_train)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	1	0	3	Braund, Mr. Owen Harris	male	22.00	1	0	A/5 21171	7.2500		S
2	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.00	1	0	PC 17599	71.2833	C85	C
3	3	1	3	Heikkinen, Miss. Laina	female	26.00	0	0	STON/O2. 3101282	7.9250		S

Fig. S1. Datos de entrenamiento (titanic_train).

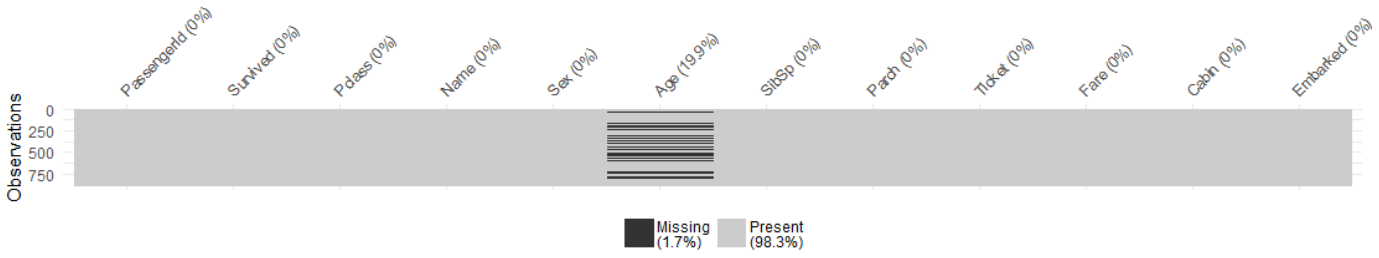


Fig. S2. Datos de entrenamiento: Porcentaje de valores faltantes por variable.

En los datos de entrenamiento, se observa un 19.9% de valores faltantes en la variable **Age**, por lo que dicha variable necesitará de la inserción de un mecanismo de imputación.

Listing 3. Titanic: Datos de evaluación.

```
head(titanic_test,3)
vis_miss(titanic_test)
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292		Q
2	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000		S
3	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875		Q

Fig. S3. Datos de evaluación (titanic_test).

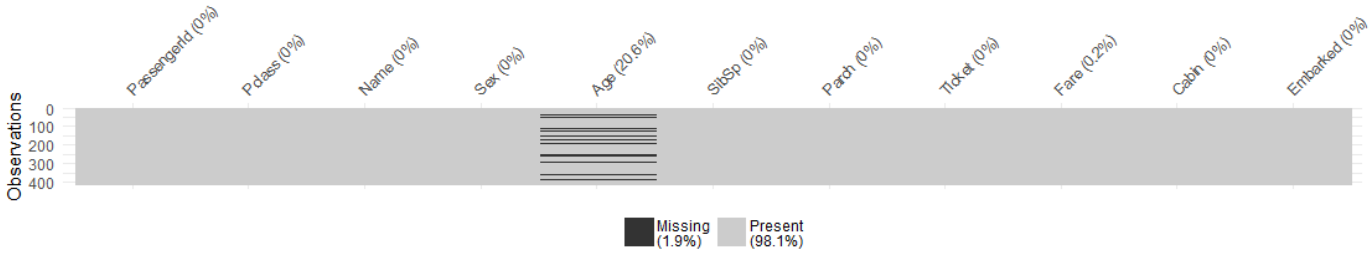


Fig. S4. Datos de evaluación: Porcentaje de valores faltantes por variable.

En los datos de evaluación, se observa un 20.6% de valores faltantes en la variable **Age**, y 0.2% en la variable **Fare**, por lo que dichas variables necesitarás de la inserción de un mecanismo de imputación.

En lo siguiente, se juntarán las bases anteriores, se transformarán a tipo factor las variables necesarias, se realizarán imputaciones sobre la variable **Age**, que será el valor medio de los datos presentes sobre los valores faltantes, se eliminará 1 dato vacío en **Fare** y 2 datos vacíos en **Embarked**. Además notamos que en los datos de evaluación no está presente la variable **Survived**, sino que ésta se encuentra en los datos titanic_gender_model por lo que se extraerá y adicionará en los datos de entrenamiento. Cabe recalcar que no se recogerán como variables predictoras aquellas que tengan una cantidad excesiva de niveles, en el caso de una variable de tipo factor, y cuando presenten una cantidad excesiva de valores faltantes.

Listing 4. Titanic: Limpieza de datos.

```
# Juntado datos
train.test <- bind_rows(titanic_train,titanic_test)

# Transformando algunas variables a factor
train.test$Survived <- as.factor(train.test$Survived)
train.test$Pclass <- as.factor(train.test$Pclass)
train.test$Name <- as.factor(train.test$Name)
train.test$Sex <- as.factor(train.test$Sex)
train.test$Ticket <- as.factor(train.test$Ticket)
train.test$Cabin <- as.factor(train.test$Cabin)
train.test$Embarked <- as.factor(train.test$Embarked)

# Valores faltantes
# Reemplazamiento en Age
mean.age <- mean(train.test$Age,na.rm = T)
train.test$Age[is.na(train.test$Age)] = mean.age

# Eliminacion en Fare
pass.na = train.test[is.na(train.test$Fare),]$PassengerId
train.test <- train.test[train.test$PassengerId != pass.na, ]

### SELECCION DE PREDICTORES
str(train.test)
train.test <- subset(train.test,select = c(2,3,5,6,7,8,10,12))

### TRAIN DATA, TEST DATA
titanic_train <- train.test[!is.na(train.test$Survived),]
titanic_train <- titanic_train[titanic_train$Embarked!= "",]
str(titanic_train)

titanic_test <- train.test[is.na(train.test$Survived),]
```

```
titanic_gender_model <- titanic_gender_model[titanic_gender_model$PassengerId != pass.na, ]
titanic_test$Survived <- as.factor(titanic_gender_model$Survived)
str(titanic_test)
```

```
> str(titanic_train)
'data.frame':   889 obs. of  8 variables:
 $ Survived: Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
 $ Pclass  : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
 $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
 $ Age     : num  22 38 26 35 35 ...
 $ SibSp   : int  1 1 0 1 0 0 0 3 0 1 ...
 $ Parch   : int  0 0 0 0 0 0 0 1 2 0 ...
 $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
> str(titanic_test)
'data.frame':   417 obs. of  8 variables:
 $ Survived: Factor w/ 2 levels "0","1": 1 2 1 1 2 1 2 1 2 1 ...
 $ Pclass  : Factor w/ 3 levels "1","2","3": 3 3 2 3 3 3 3 2 3 3 ...
 $ Sex     : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
 $ Age     : num  34.5 47 62 27 22 14 30 26 18 21 ...
 $ SibSp   : int  0 1 0 0 1 0 0 1 0 2 ...
 $ Parch   : int  0 0 0 0 1 0 0 1 0 0 ...
 $ Fare    : num  7.83 7 9.69 8.66 12.29 ...
 $ Embarked: Factor w/ 4 levels "","C","Q","S": 3 4 3 4 4 4 3 4 2 4 ...
```

Fig. S5. Titanic: Tipo de Variables.

Luego de haberse realizado la imputación y eliminación de ciertos datos innecesarios, se cuentan con 889 datos de entrenamiento y 417 datos de evaluación, ambas con 8 variables, en donde la variable **Survived** será la variable respuesta, representándose de la siguiente forma

	Survived
Sobrevivió	1
No Sobrevivió	0

Nota: Cabe recalcar que la variable Embarked solo cuenta con 3 factores, sin embargo en FIG S5., se indica la existencia de 4, esto se debe a que al inicio existía la presencia de un factor vacío, que luego fueron eliminados dado que eran solo 2 datos que contaban con ello.

ANÁLISIS DESCRIPTIVO.

Antes de iniciar con ciertos análisis entre las variables, se diseñarán funciones que ayudarán en la representación gráfica.

Listing 5. Análisis Descriptivo: Funciones

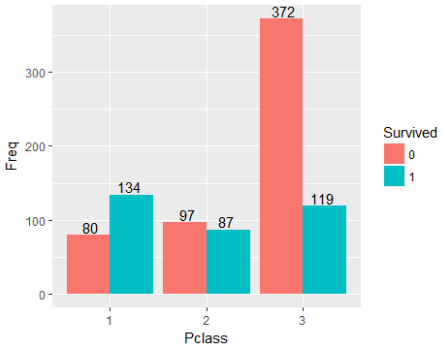
```
bar_function<-function(x){
        a<-as.data.frame(table(titanic_train[ ,x],factor(titanic_train[,"Survived"])))
        ggplot(a,aes(Var1,Freq,fill=Var2))+
                geom_bar(stat="identity",position ="dodge")+
                geom_text(aes(x=Var1,y=Freq,label=Freq),position = position_dodge(width = .8),
                    vjust=-.2)+
                labs(x=x,y="Freq",fill="Survived")
}

my_mosaic<-function(file,xcolname,ycolname){
        file<-file[,c(xcolname,ycolname)]
        cname<-c(xcolname,ycolname)
        a<-file[colnames(file)%in%cname]
        xname<-as.name(xcolname)
        yname<-as.name(ycolname)
        prob<-signif(prop.table(table(a),1),digits = 2)
        mosaic(prob,pop=F,shade = F,legend=T,rot_lables=c(0,90,0,0),
                labeling_args = list(set_varnames=c(xname=xcolname,yname=ycolname)),
                main = "Survived Rate")
                labeling_cells(text = prob,margin = 0)(prob)
}
```

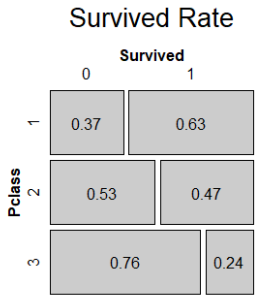
Habiéndose diseñado las funciones, se procede a los siguiente análisis

Listing 6. Pclass - Survived

```
bar_function("Pclass")
my_mosaic(titanic_train,"Pclass","Survived")
```



(a) Pclass (Frecuencia) - Survived



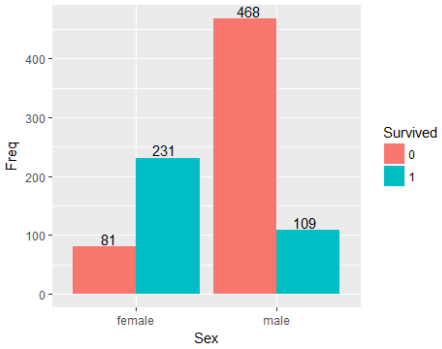
(b) Pclass (Porcentaje) - Survived

Fig. S6. Pclass (Clase).

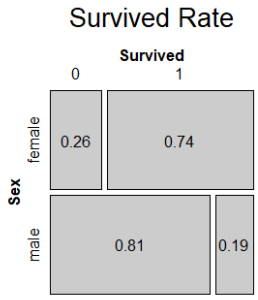
Interpretación: La mayoría de pasajeros viajaron en clase 3, en donde se registró la tasa de sobrevivencia más baja (0.24). A diferencia de los que los que viajaron en clase 1, que siendo una cantidad menor, la tasa de sobrevivencia fue la más alta (0.63).

Listing 7. Sex - Survived

```
bar_function("Sex")
my_mosaic(titanic_train,"Sex","Survived")
```



(a) Sex (Frecuencia) - Survived



(b) Sex (Porcentaje) - Survived

Fig. S7. Sex (Sexo).

Interpretación: Se registró una tasa de sobrevivencia mucho más alta en mujeres (0.74) que en hombres (0.19).

Listing 8. Age - Survived.

```
ggplot(titanic_train,aes(as.factor(Survived),Age))+
        geom_violin(aes(fill=Survived))+
        labs(x="Survived")+
        geom_hline(aes(yintercept=10),lty=2,lwd=1,col="red")+
        scale_y_continuous(breaks = seq(0,80,10))+
        theme(legend.position = "none")
```

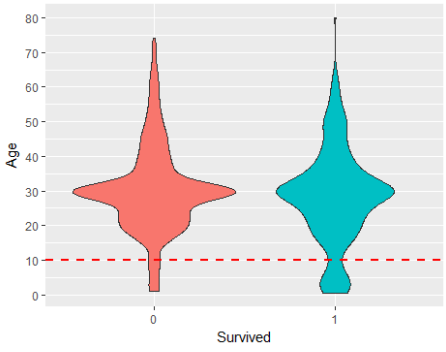


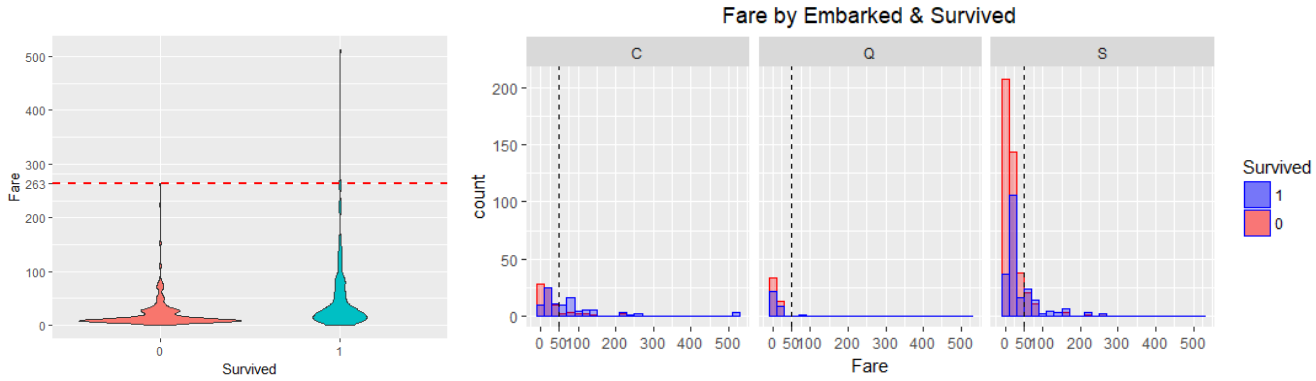
Fig. S8. Age - Survived.

Interpretación: Se realizó un corte con el objetivo de observar pasajeros de edad baja (< 10 años). Se observó que pasajeros con edad menor que 10 años tienen alta probabilidad de sobrevivir.

Listing 9. Fare - Survived.

```
ggplot(titanic_train,aes(as.factor(Survived),Fare))+
        geom_violin(aes(fill=Survived))+ labs(x="Survived")+
        geom_hline(aes(yintercept=max(titanic_train[titanic_train$Survived==0,]$Fare)),lty=2,
                   lwd=1,col="red")+
        scale_y_continuous(breaks=c(seq(0,200,100),
                            max(titanic_train[titanic_train$Survived==0,]$Fare),
                            seq(300,500,100)))+
        theme(legend.position = "none")

ggplot(titanic_train,aes(Fare))+
        geom_histogram(data=titanic_train[titanic_train$Survived==0,],
                       aes(fill="red"),colour="red",binwidth = 20,alpha=.3)+
        geom_histogram(data=titanic_train[titanic_train$Survived==1,],
                         aes(fill="blue"),colour="blue",binwidth = 20,alpha=.3)+
        geom_vline(aes(xintercept=50),lty=2,lwd=.5)+scale_colour_manual(name="Survived",
                       values = c("red"="red","blue"="blue"),labels=c("red"=0,"blue"=1))+
        scale_fill_manual(name="Survived",
                          values = c("red"="red","blue"="blue"),labels=c("red"=0,"blue"=1))+
        scale_x_continuous(breaks = c(0,50,seq(100,500,100)))+
        labs(title="Fare by Embarked & Survived")+
        theme(plot.title = element_text(hjust = .5))+facet_grid(.~Embarked)
```



(a) Fare - Survived

(b) Fare - Embarked - Survived

Fig. S9. Fare (Tarifa).

Interpretación: En general, pasajeros que usaron tarifas más altas, tuvieron más chances de sobrevivir. Además se puede observar que en el embarque S, se compraron más tarifas por debajo de 50 que en cualquier otro embarque, trayendo consigo una alta frecuencia de pasajeros que no sobrevivieron.

MODELO LOGÍSTICO

Se realiza un modelo logístico usando todas las variables predictoras, indicadas en **Fig. S5.**.

Listing 10. Modelo Logístico

```
mod<- glm(Survived ~ .,family = binomial,
          data = titanic_train)
summary(mod)

# MATRIZ DE CONFUSION
p = predict(mod,
            newdata = subset(titanic_test,
            select = c(2:8)),
            type = 'response')
p <- ifelse(p > 0.5,1,0)
table(clase_predicha = p,
      clase_real = titanic_test$Survived)
c(mean(p == titanic_test$Survived),
  mean(p != titanic_test$Survived))
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.106628   0.476646   8.616  < 2e-16 ***
Pclass2     -0.925239   0.297932  -3.106  0.00190 **
Pclass3     -2.150054   0.297752  -7.221 5.16e-13 ***
Sexmale     -2.709536   0.201347 -13.457  < 2e-16 ***
Age         -0.039367   0.007889  -4.990 6.02e-07 ***
SibSp       -0.322293   0.109595  -2.941  0.00327 **
Parch       -0.095458   0.119045  -0.802  0.42263
Fare         0.002257   0.002462   0.917  0.35936
EmbarkedQ   -0.026843   0.381586  -0.070  0.94392
EmbarkedS   -0.446383   0.239749  -1.862  0.06262 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Fig. S10. Modelo Logístico: Coeficientes y significancia.

Matriz de Confusión	Clase Real	
Clase predicha	0	1
0	252	10
1	13	142

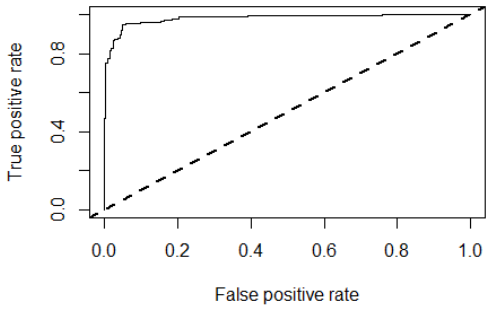
El modelo ha sido capaz de predecir correctamente el 94.48% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 5.52%.

Se graficará la curva ROC y se observará el AUC (área bajo la curva) obtenido mediante este modelo de clasificación.

Listing 11. Curva ROC

```
p = predict(mod,
            newdata = subset(titanic_test, select = c(2:8)),
            type = 'response')
pr = prediction(p, titanic_test$Survived)
prf = performance(pr, measure = 'tpr', x.measure = 'fpr')
plot(prf)
abline(0,1, lwd = 2, lty = 2)

auc = performance(pr,measure='auc')
str(auc)
```



Se tiene una AUC de 0.98, esto indica que el modelo logístico diseñado es muy cercano a un clasificador perfecto.

Fig. S11. Curva ROC.

MODELO LOGÍSTICO 2

En el modelo logístico previo, se observó que con respecto a la variable **Embarked** se presentó un p-valor alto con respecto al nivel **EmbarkedQ** y un p-valor relativemante bajo para el nivel **EmbarkedS**. En lo siguiente, se realizará un *Test Wald* para evaluar si estadísticamente existen diferencias significativas con respecto a los niveles de la variable **Embarked**.

Listing 12. Modelo Logístico

```
wald.test(b = coef(mod), Sigma = vcov(mod), Terms = 9:10)
```

```
Wald test:
----------

Chi-squared test:
X2 = 4.5, df = 2, P(> X2) = 0.11
```

Como se observa, se obtiene un p-valor de 0.11 en el test de Wald, esto indica que no existen diferencias significativas entre los niveles de la variable **Embarked**.

Fig. S12. Test de Wald sobre Variable Embarked del Modelo Logístico previo.

Dicho esto se diseñará un modelo logístico sin la variable **Embarked**, de la siguiente forma

Listing 13. Modelo Logístico 2

```
mod2<- glm(Survived ~ .-Embarked,family = binomial,
           data = titanic_train)
           summary(modelo.logistico)

# MATRIZ DE CONFUSION
p = predict(mod2,
            newdata = subset(titanic_test,
            select = c(2:8)),
            type = 'response')
p <- ifelse(p > 0.5,1,0)
table(clase_predicha = p,
clase_real = titanic_test$Survived)
c(mean(p == titanic_test$Survived),
  mean(p != titanic_test$Survived))
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.838856   0.446787   8.592  < 2e-16 ***
Pclass2     -1.018940   0.293920  -3.467 0.000527 ***
Pclass3     -2.144427   0.289514  -7.407 1.29e-13 ***
Sexmale     -2.753782   0.199476 -13.805  < 2e-16 ***
Age         -0.039706   0.007857  -5.054 4.33e-07 ***
SibSp       -0.349395   0.109554  -3.189 0.001426 **
Parch       -0.112362   0.117611  -0.955 0.339389
Fare         0.002966   0.002441   1.215 0.224262
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Fig. S13. Modelo Logístico 2: Coeficientes y significancia.

Matriz de Confusión	Clase Real	
Clase predicha	0	1
0	252	9
1	13	143

El modelo ha sido capaz de predecir correctamente el 94.72% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 5.28%. Existe una mejor mínima con respecto al modelo logístico anterior.

Se graficará la curva ROC y se observará el AUC (área bajo la curva) obtenido mediante este modelo de clasificación.

Listing 14. Curva ROC

```
p = predict(mod2,
            newdata = subset(titanic_test, select = c(2:8)),
            type = 'response')
pr = prediction(p, titanic_test$Survived)
prf = performance(pr, measure = 'tpr', x.measure = 'fpr')
plot(prf)
abline(0,1, lwd = 2, lty = 2)

auc = performance(pr,measure='auc')
str(auc)
```

Se tiene una AUC de 0.983, esto indica que el modelo logístico diseñado es muy cercano a un clasificador perfecto, aún mejor que el modelo logístico previo.

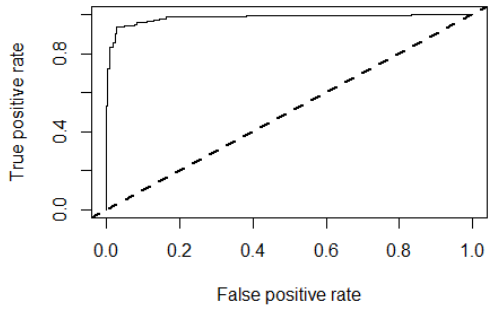


Fig. S14. Curva ROC.

MODELO SVM

Se diseñaran modelos SVM con distintos parámetros C y gamma. Además de visualizarse los errores de clasificación por parte de todos los modelos presentes.

Listing 15. Modelos SVM

```
modelo_svm <- tune.svm(Survived~ . -Survived,data = titanic_train, kernel = "radial",
                       gamma = c(0.01,0.1,0.5,1,4,8), cost = c(0.01,0.1,0.5,1,4,8))

ggplot(data = modelo_svm$performances, aes(x = cost, y = error, color = as.factor(gamma))) +
        geom_line() + geom_point() +
        labs(title = "Error de clasificacion vs hiperparametros C y gamma", color = "gamma") +
        theme_bw() + theme(legend.position = "bottom")
```

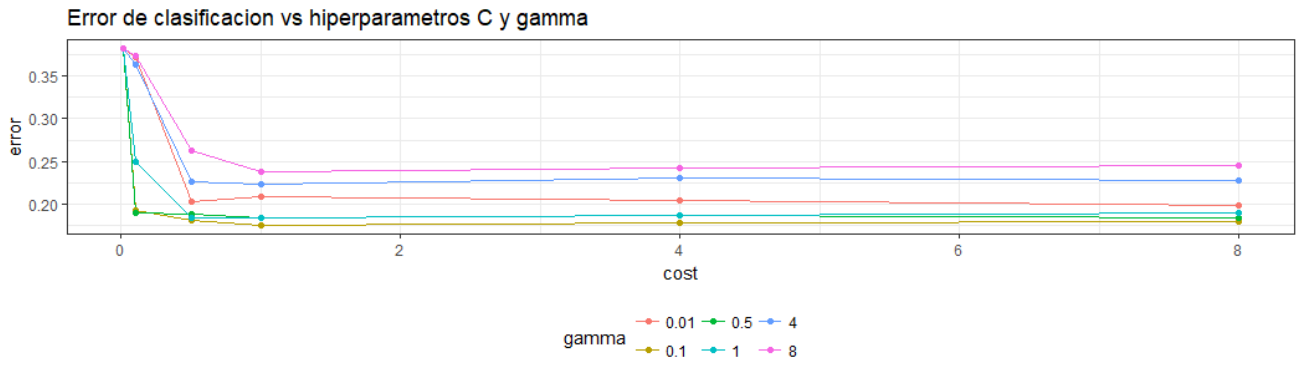


Fig. S15. Modelos SVM con diferentes parámetros de diseño.

Posteriormente, se recoge el mejor modelo

Listing 16. Mejor modelo SVM

```
modelo_svm$best.parameters
modelo_svm_mejor <- modelo_svm$best.model
```

gamma	cost
0.1	1

Listing 17. Matriz de confusión

```
svm.predict <- predict(modelo_svm_mejor, titanic_test[,2:8])

table(clase_predicha = svm.predict,clase_real = titanic_test$Survived)
paste("% de acierto:", mean(svm.predict == titanic_test$Survived))
paste("% de error:", mean(svm.predict != titanic_test$Survived))
```

Matriz de Confusión	Clase Real	
Clase predicha	0	1
0	252	9
1	13	143

El modelo ha sido capaz de predecir correctamente el 94.72% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 5.28%. Resultados similares al modelo logístico 2.

RANDOM FOREST

Se diseñaran un modelo Random Forest usando todas las variables predictoras.

Listing 18. Modelo Random Forest

```
set.seed(604)
mod.rf <- randomForest(Survived ~ .,data = titanic_train)

plot(mod.rf, ylim=c(0,0.36))
legend('topright', colnames(mod.rf$err.rate), col=1:3, fill=1:3)
```

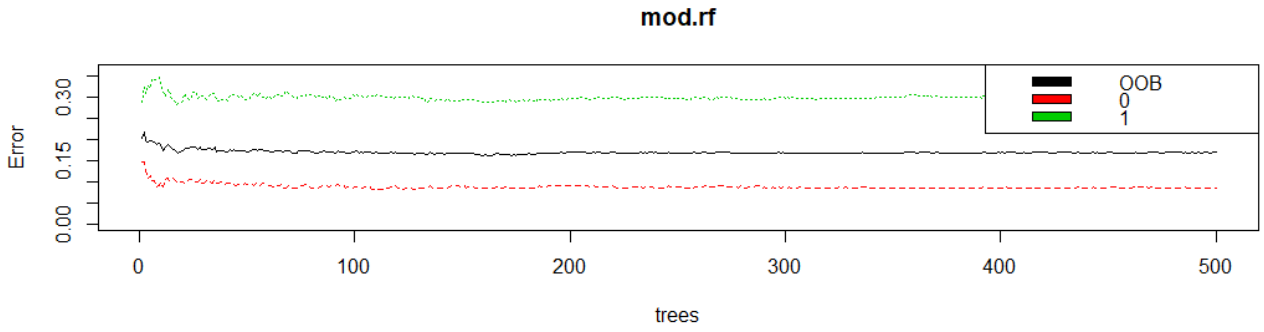


Fig. S16. OOB: Tasa de error medio cercano al 15%. **0**: Tasa de error para los no sobrevivientes (**Survived**=0) cercano a 0.10%. **1**: Tasa de error para los sobrevivientes (**Survived**=1) cercano a 0.30%.

Listing 19. Matriz de confusión

```
p = predict(mod.rf, titanic_test)
table(clase_predicha = p,clase_real = titanic_test$Survived)
c(mean(p == titanic_test$Survived),mean(p != titanic_test$Survived))
```

Matriz de Confusión	Clase Real	
Clase predicha	0	1
0	247	32
1	18	120

El modelo ha sido capaz de predecir correctamente el 88% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 12%.

RANDOM FOREST 2

En el diseño de este modelo 2 Random Forest, primero se evaluarán la importancia de las variables en el modelo 1 Random Forest, con lo que posteriormente el diseño se dará sobre las variables con mayor importancia

Listing 20. Importancia de las variables

```
importance    <- importance(mod.rf)
varImportance <- data.frame(Variables = row.names(importance),
                            Importance = round(importance[ ,'
                                MeanDecreaseGini'],2))

rankImportance <- varImportance %>%
        mutate(Rank = paste0('#',dense_rank(desc(Importance))))

ggplot(rankImportance, aes(x = reorder(Variables, Importance),
                           y = Importance, fill = Importance))
                          +
        geom_bar(stat='identity') +
        geom_text(aes(x = Variables, y = 0.5, label = Rank),
                  hjust=0, vjust=0.55, size = 4,
                  colour = 'red') +
        labs(x = 'Variables') +
        coord_flip() +
        theme_few()
```

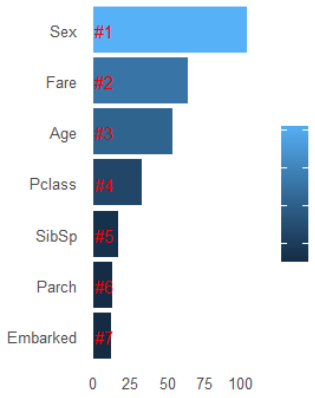


Fig. S17. Importancia de las variables en el Modelo 1 Random Forest.

Se diseñaran un modelo Random Forest sin adición de las variables menos importantes (**Embarked**, **Parch**, **SibSp**).

Listing 21. Modelo 2 Random Forest

```
set.seed(604)
mod.rf <- randomForest(Survived ~ . -Embarked-Parch-SibSp,data = titanic_train)

plot(mod.rf, ylim=c(0,0.36))
legend('topright', colnames(mod.rf$err.rate), col=1:3, fill=1:3)
```

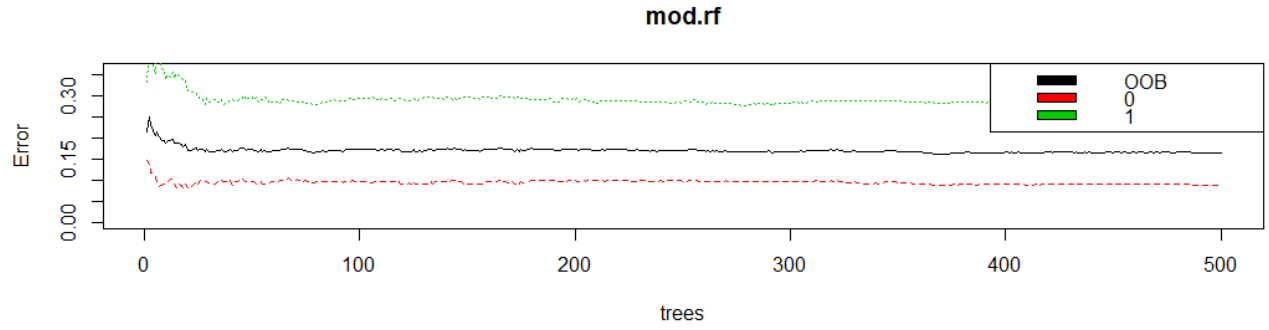


Fig. S18. OOB: Tasa de error medio cercano al 15%. **0**: Tasa de error para los no sobrevivientes (**Survived**=0) cercano a 0.10%. **1**: Tasa de error para los sobrevivientes (**Survived**=1) cercano a 0.30%.

Listing 22. Matriz de confusión

```
p = predict(mod.rf, titanic_test)
table(clase_predicha = p,clase_real = titanic_test$Survived)
c(mean(p == titanic_test$Survived),mean(p != titanic_test$Survived))
```

Matriz de Confusión	Clase Real	
Clase predicha	0	1
0	252	28
1	13	124

El modelo ha sido capaz de predecir correctamente el 90.2% de las observaciones, mejor de lo que cabría esperar por azar (50%). El *test error* es de 9.8%. Resultados similares al modelo 2 Random Forest.

CONCLUSIÓN

De los modelos estudiados, quienes menor *test error* presentan son el modelo Logístico 2 y el modelo SVM con hiperparámetros gamma = 0.1 y C=1.