Trabajo final

Marcos Gómez de Quero Santos, Pablo López Martínez

12/31/2020

Abstract

Clasificación muestras de cáncer de mama. El problema trata de clasificar muestras de cáncer de mama atendiendo a varias características medidas. Los tumores han sido asignados por expertos humanos a dos clases, “benigno” y “maligno”. Después de leer el artículo [2] disponible en la intranet, implementar el sistema difuso que permita clasificar las muestras tumorales. Comentar las ventajas e inconvenientes del sistema difuso diseñado y la posible interpretación de los resultados en forma de reglas. Para nota: Implementar el método del gradiente descendente para estimar los parámetros óptimos del sistema difuso a partir de un conjunto de entrenamiento. Comparar los resultados con los que se obtienen con el sistema difuso obtenido de manera heurística y con un sistema neurodifuso como por ej. ANFIS.

Introducción

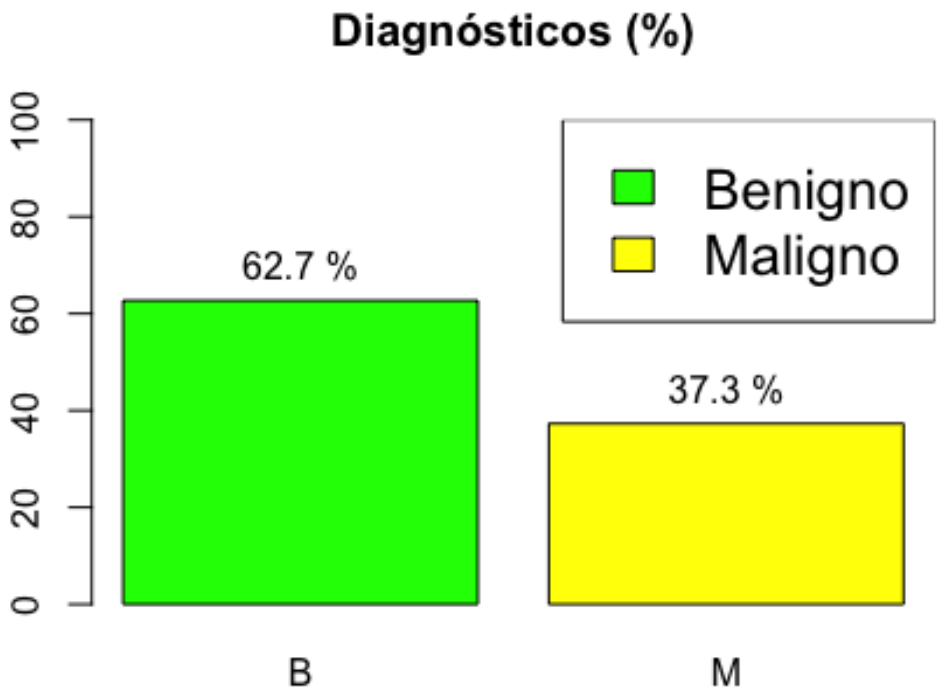
Elaboraremos un diagnóstico dividiendo entre Maligno o M y Benigno o B. Utilizaremos algoritmos de aprendizaje automático que se servirán de un extenso dataset con 32 características distintas.

##Analisis y Desarrollo

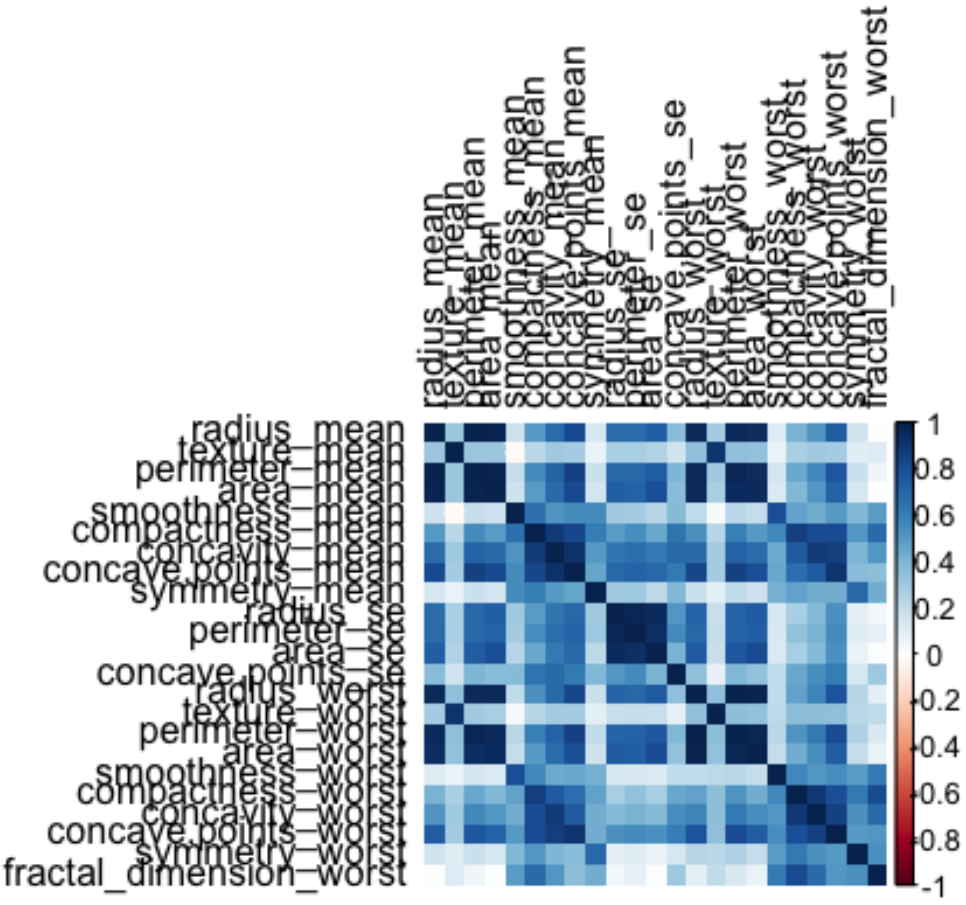
Importamos el dataset con el contenido de las observaciones

```
setwd("~/Downloads")
getwd()
datos_cancer <-read.csv("data.csv")

datos_cancer[!complete.cases(datos_cancer)];

## Borramos la columna de identificadores, ya que esta columna no se
usará
datos_cancer<-datos_cancer[-1];
## Conversión a 0 y 1 respectivamente Benigno y Maligno
datos_cancer$diagnosis<-factor(datos_cancer$diagnosis);

porcentaje <- round(prop.table(table(datos_cancer$diagnosis))*100,digits
= 1)
indicador  <-barplot(porcentaje,main="Diagnósticos (%)",
col=c("green","yellow"),ylim=c(0,100))

legend("topright", c("Benigno","Maligno"), cex=1.5,
```

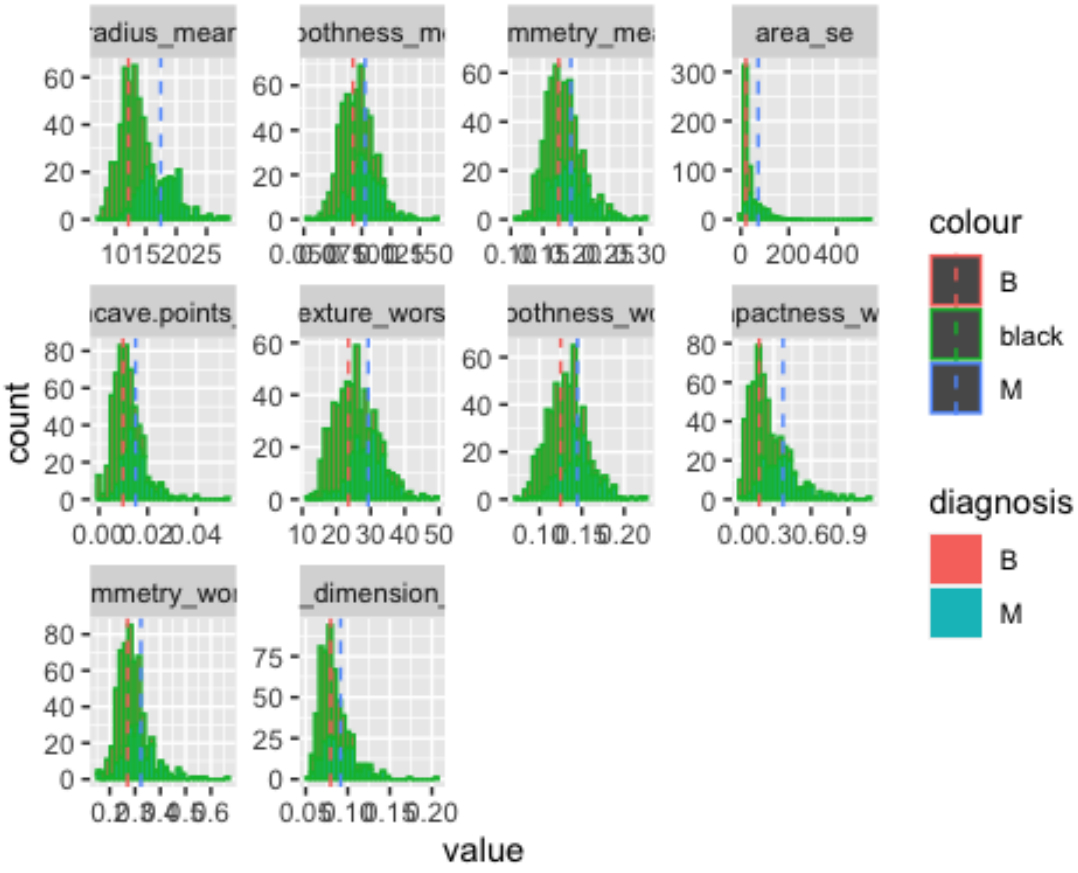
```
fill=c("green","yellow"))
text(x = indicador, y = porcentaje, label = paste(porcentaje,"%"), pos =
3, cex = 1)
```



Tratamiento de los vectores de correlación y datos en tabla de correlación



Tratamiento de los vectores de correlación y resumen de los datos



Retiramos de los datos observados anteriormente aquellos que tienen medias cercanas aplicando un factor de 0.05. De esta manera reducimos el numero de caracteristicas para el entrenamiento de los modelos de aprendizaje.

Dividimos los datos en dos conjuntos, un conjunto para pruebas y un conjunto para el entrenamiento. Utilizaremos el 80% de los elementos para el entrenamiento y el resto se usará para las pruebas.

Se utilizarán los siguientes algoritmos para la clasificación: -Random Forest -GLM

```
#Eliminar con media aproximada a los dos tipos de datos
diffmean<-ddply(mu, ~variable, summarise, difmean=diff(grp.mean));
datos_cancer = data.frame(diagnosis=datos_cancer[,1],
datos_cancer[which(diffmean$difmean>0.05)+1]);



#Preparamos los datos de prueba
splitdata <- function(df,n){
  nd = nrow(df)
  l  = round(n*nd/100)
  trainind <- sample(seq_len(nd),size = l)
  datos_cancer_train <- df[trainind,]
```

```
  datos_cancer_test  <- df[-trainind,]
  data = list(datos_cancer_train, datos_cancer_test)
  return(data)
}

#Dividimos los datos en dos subconjuntos, unos son de entrenamiento y
otros son de prueba
data = splitdata(datos_cancer,80);
datos_cancer_train = data[[1]];
datos_cancer_test  = data[[2]];

#Utilizamos validación cruzada:
fitControl <- trainControl(method = "repeatedcv",
                           number = 10,
                           repeats = 2,
                           savePredictions="final",
                           preProcOptions = list(thresh = 0.99),
                           classProbs = TRUE)
```

Utilización del algoritmo GLM

Es una generalización flexible de la regresión lineal ordinaria que permite variables de respuesta que tienen modelos de distribución de errores distintos de una distribución normal.

```
#Algoritmo GLM (Logistic regression multiclass)
fit_glmnet <- train (diagnosis~.,
                     datos_cancer_train,
                     method = "glmnet",
                     tuneLength = 20,
                     metric = "Accuracy",
                     preProc =c ("center", "scale"),
                     trControl = fitControl)


pred_glmnet <- predict(fit_glmnet, datos_cancer_test);
confusionMatrix(pred_glmnet, datos_cancer_test$diagnosis);

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 68  2
##          M  1 43
##
##                Accuracy : 0.9737
##                  95% CI : (0.925, 0.9945)
##     No Information Rate : 0.6053
##     P-Value [Acc > NIR] : <2e-16
##
```

```
##                   Kappa : 0.9447
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9855
##             Specificity : 0.9556
##          Pos Pred Value : 0.9714
##          Neg Pred Value : 0.9773
##              Prevalence : 0.6053
##          Detection Rate : 0.5965
##    Detection Prevalence : 0.6140
##       Balanced Accuracy : 0.9705
##
##        'Positive' Class : B
##
```

```
CrossTable(x=datos_cancer_test$diagnosis,y=pred_glmnet,prop.chisq =
FALSE,prop.tbl=FALSE,prop.col=FALSE,prop.row=FALSE);
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  114
##
##
##                              | pred_glmnet
## datos_cancer_test$diagnosis |         B |         M | Row Total |
## ----------------------------|-----------|-----------|-----------|
##                           B |        68 |         1 |        69 |
##                             |     0.986 |     0.014 |     0.605 |
##                             |     0.971 |     0.023 |           |
##                             |     0.596 |     0.009 |           |
## ----------------------------|-----------|-----------|-----------|
##                           M |         2 |        43 |        45 |
##                             |     0.044 |     0.956 |     0.395 |
##                             |     0.029 |     0.977 |           |
##                             |     0.018 |     0.377 |           |
## ----------------------------|-----------|-----------|-----------|
##                Column Total |        70 |        44 |       114 |
##                             |     0.614 |     0.386 |           |
## ----------------------------|-----------|-----------|-----------|
```

```
##
##

d <- ifelse(datos_cancer_test$diagnosis == "M",1,0);
pd_glm <- ifelse(pred_glmnet == "M",1,0);
roc_obj <- roc(d,pd_glm);

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

## Setting direction: controls < cases

print(paste0("Área bajo la curva: ",auc(d,pd_glm)));

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

## [1] "Área bajo la curva: 0.970531400966184"
```

Utilización del algoritmo Random Forest

Combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. Cada arbol revela una predicción de clase y la clase más votada se convierte en la predicción del modelo.

```
fit_rf <- train (diagnosis~.,
                 datos_cancer_train,
                 method = "ranger",
                 metric = "Mean_F1",
                 preProc = c("center", "scale"),
                 trControl = fitControl)

## Warning in train.default(x, y, weights = w, ...): The metric "Mean_F1"
was not
## in the result set. Accuracy will be used instead.

pred_rf <- predict(fit_rf, datos_cancer_test);
confusionMatrix(pred_rf, datos_cancer_test$diagnosis);

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 67  2
##          M  2 43
##
##                Accuracy : 0.9649
##                  95% CI : (0.9126, 0.9904)
##     No Information Rate : 0.6053
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9266
```

```
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9710
##             Specificity : 0.9556
##          Pos Pred Value : 0.9710
##          Neg Pred Value : 0.9556
##              Prevalence : 0.6053
##          Detection Rate : 0.5877
##    Detection Prevalence : 0.6053
##       Balanced Accuracy : 0.9633
##
##        'Positive' Class : B
##
```

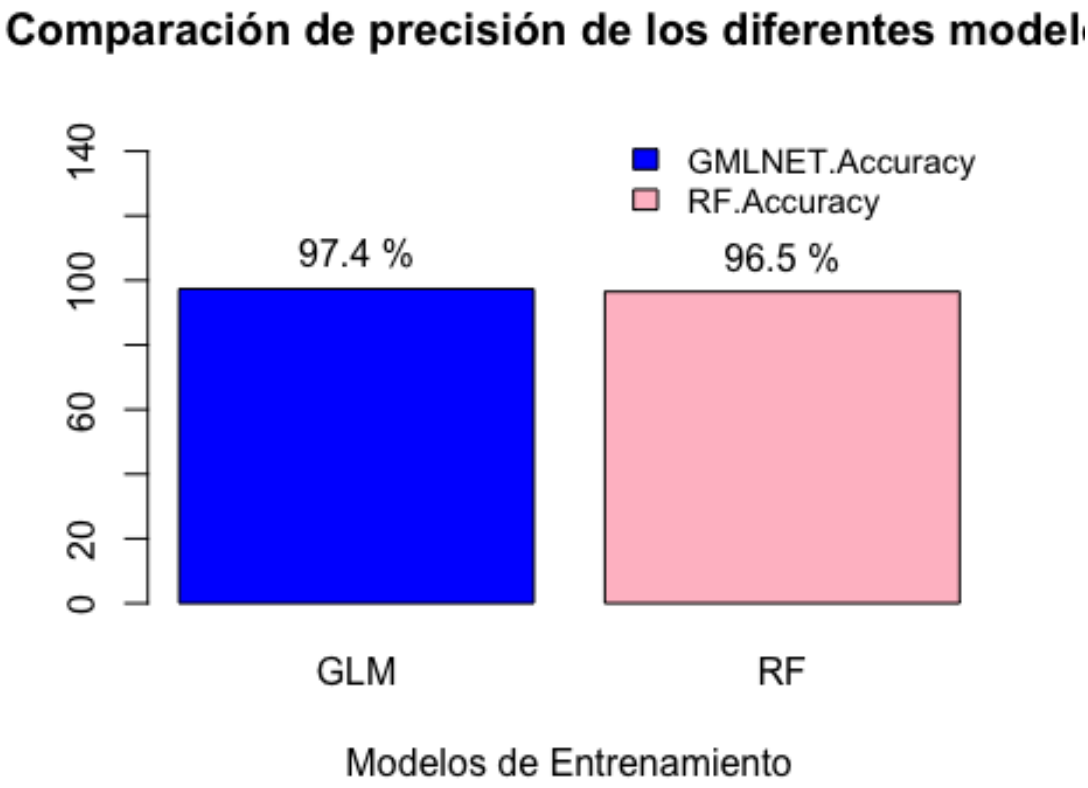
```
CrossTable(x=datos_cancer_test$diagnosis, y=pred_rf, prop.chisq = FALSE,
prop.tbl=FALSE, prop.col=FALSE, prop.row=FALSE);
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  114
##
##
##                              | pred_rf
## datos_cancer_test$diagnosis |         B |         M | Row Total |
## ----------------------------|-----------|-----------|-----------|
##                           B |        67 |         2 |        69 |
##                             |     0.971 |     0.029 |     0.605 |
##                             |     0.971 |     0.044 |           |
##                             |     0.588 |     0.018 |           |
## ----------------------------|-----------|-----------|-----------|
##                           M |         2 |        43 |        45 |
##                             |     0.044 |     0.956 |     0.395 |
##                             |     0.029 |     0.956 |           |
##                             |     0.018 |     0.377 |           |
## ----------------------------|-----------|-----------|-----------|
##                Column Total |        69 |        45 |       114 |
##                             |     0.605 |     0.395 |           |
## ----------------------------|-----------|-----------|-----------|
##
##
```

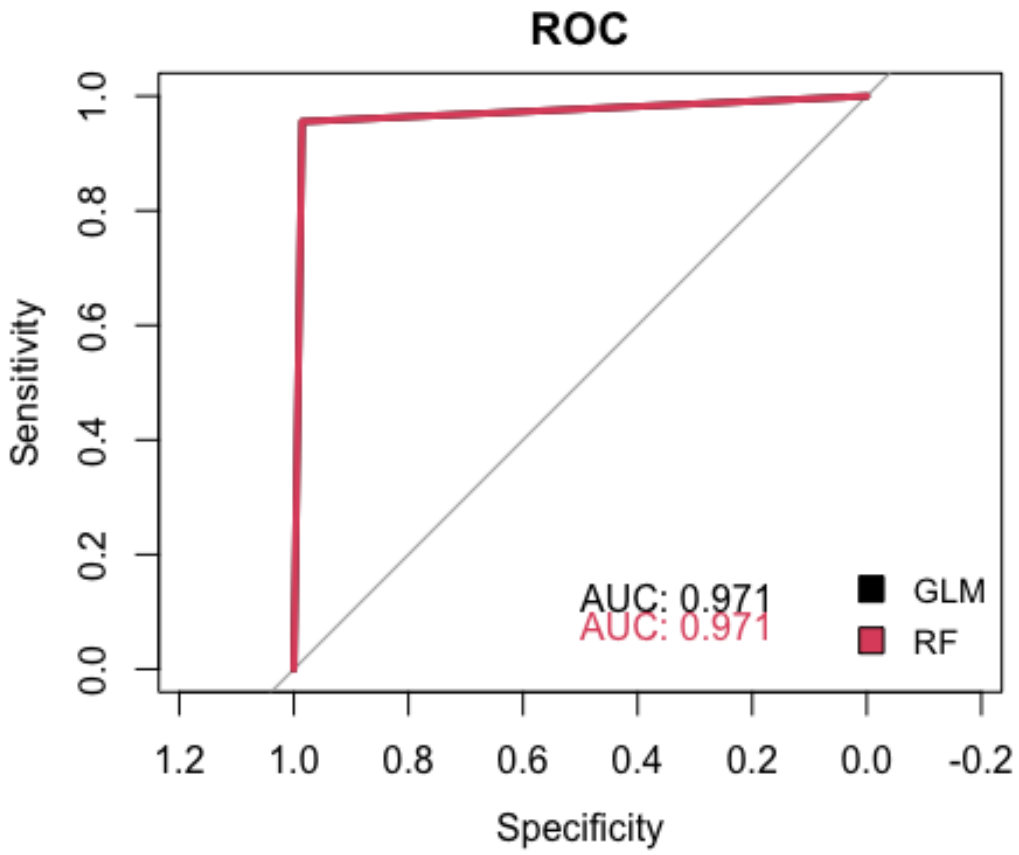
```
pd_rf <- ifelse(pred_glmnet == "M",1,0);
roc_obj <- roc(d,pd_rf);

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

## Setting direction: controls < cases

print(paste0("Área bajo la curva: ",auc(d,pd_rf)));

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

## [1] "Área bajo la curva: 0.970531400966184"
```

Presentamos los resultados finales de nuestro estudio

```
model_list <- list(GMLNET=fit_glmnet, RF = fit_rf);
resamples <- resamples(model_list);

model_confmat <- lapply(model_list, function(x)
confusionMatrix(predict(x, datos_cancer_test),
datos_cancer_test$diagnosis));
model_accuracy <- sapply(model_confmat, function(x)
x$overall['Accuracy']);


#Diagrama de precisión del modelo

xx<-barplot(100*model_accuracy,main="Comparación de precisión de los
diferentes modelos",ylim=c(0,150),xlab = "Modelos de Entrenamiento",
names.arg =c("GLM","RF"),  col=c("blue","pink"));
legend("topright", c(names(model_accuracy)), cex=0.9, bty="n",
fill=c("blue","pink"))
text(x = xx, y = 100*model_accuracy, label =
paste(100*round(model_accuracy,3),"%"), pos = 3, cex =1, col = "black");
```




```
#Curva ROC
plot(roc(d, pd_glm),
     col=1, lwd=3,print.auc.y = .15,print.auc=TRUE, main="ROC");

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

## Setting direction: controls < cases

plot(roc(d, pd_rf),
     col=2, lwd=3,print.auc.y = .10,print.auc=TRUE,add=TRUE);

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

legend("bottomright",c("GLM","RF"),cex=0.9, bty="n", fill=1:3)
```



Conclusiones

Tras ejecutar el programa múltiples veces, hemos llegado a la conclusión de que el algoritmo con mayor precisión es el algoritmo GLM.

Como se puede observar, hemos utilizado dos modelos de predicción diferentes. Con los cuales hemos obtenido datos sobre el cáncer de mama.

La mejora en las predicciones mejorará con la mejora en la precisión de la clasificación.

En esta práctica, hemos utilizado dos algoritmos diferentes con los cuales hemos obtenido una precisión muy elevada en sus predicciones.

Sinceramente, desde nuestro punto de vista, ha sido una práctica muy complicada, con la que hemos tenido que buscar mucha información, ayuda y apoyo de otras personas. Ya que nos hemos apoyado bastante en muchas cosas, nos hemos dado cuenta de que el trabajo en equipo y la búsqueda de información es muy importante en esta clase de proyectos. También hemos visto cómo utilizar estos algoritmos, los cuales son muy útiles para estudios en la vida real en diferentes aspectos y campos. A parte de haber visto ya la heurística en otra asignatura, Investigación Operativa, aquí hemos afianzado conceptos y nuestra perspectiva sobre los algoritmos ha cambiado de tal forma que los vemos como algo bastante importante.

Bibliografía

https://www.jmlr.org/papers/volume13/biau12a/biau12a.pdf

https://aprendeia.com/aprendizaje-supervisado-random-forest-classification/#:~:text=Random%20Forest%20es%20un%20m%C3%A9todo,de%20regresi%C3%B3n%20como%20de%20clasificaci%C3%B3n.&text=Para%20clasificar%20un%20nuevo%20objeto,es%20la%20predicci%C3%B3n%20del%20algoritmo

https://www.kaggle.com/uciml/breast-cancer-wisconsin-data

http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/Categor/Tema3Cate.pdf

https://towardsdatascience.com/building-a-simple-machine-learning-model-on-breast-cancer-data-eca4b3b99fa3

https://es.wikipedia.org/wiki/Random_forest

http://halweb.uc3m.es/esp/Personal/personas/durban/esp/web/GLM/curso_GLM.pdf

https://www.uv.es/antoniol/EEEMA/Textos/glm.pdf

https://www.math.mcgill.ca/yyang/resources/doc/randomforest.pdf