INTRODUCCION AL ANALISIS MULTIVARIADO

Lab. No.5 - INDICADORES DE DESEMPEÑO

1. Cargue los paquetes `caret`, `rpart`, `rattle`, `DT`, `ROCR` y `plotly`.
2. Cargue la base de Credit. Declare la variable sobreendeudado como factor. Divida el archivo de datos de base2: uno para entrenar llamado `train` (80%) y el otro para predecir llamado `test` (20%). Use la instrucción `RNGkind(sample.kind = "Rounding")` y semilla igual a 10.

- Haga un árbol de decisión con la base de entrenamiento (`mod1`). Obtenga la clasificación de la base de validación y llámele `pred1`. Recuerde que en el predict debe usar `type="class"`.
- Haga un modelo de regresión logística con la base de entrenamiento y todas las variables (`mod2`). Obtenga la clasificación de la base de validación (recuerde que en el predict debe usar `type="response"` para obtener la probabilidad de éxito y luego asignar aquellos que tienen una probabilidad mayor a 0.5 al grupo 1), llámele `pred2`.
- Compare las dos clasificaciones.
- Obtenga la matriz de distancias entre los elementos de la base de entrenamiento y la base de validación. Use la distancia de Gower en la función `daisy` en la librería `cluster`. Pegue primero las dos bases y obtenga la matriz de distancias de todos contra todos, luego extraiga solo la parte que compara los elementos de entrenamiento contra los de validación.
- Haga la clasificación de la base de validación con k=5 vecinos más cercanos y llámele `pred3`.
- Compare esta clasificación con las 2 anteriores.

3. Desarrolle un función en R llamada `eval` que le permita calcular los indicadores de desempeño (`e,FP y FNC`) derivados de la matriz de confusión para un modelo de clasificación de dos clases. La función debe recibir la variable respuesta de la base de validación y la clasificación de esa misma base obtenida con cualquier método.

- Ejecute la función para el modelo generado con el árbol de decisión y la clasificación de la base de validación (`pred1`).
- Ejecute la función para el modelo generado con la regresión logística y la clasificación de la base de validación (`pred2`).
- Ejecute la función para el modelo generado con 5 vecinos más cercanos y la clasificación de la base de validación (`pred3`).
- Haga una tabla con los resultados de los 3 modelos.

6. Use la función `prediction` de la librería `ROCR` para obtener los elementos para hacer la Curva ROC y obtener el AUC. Debe dar dos variables para esta función, primero la predicción en forma numérica y el vector de respuesta: `prediction(pred,y)`. Aplique la función con el primer modelo, ponga el resultado en `predict1`.

- Extraiga el auc con `attributes(performance(predict1,"auc"))$y.values[[1]]*100`.

- Para extraer los falsos positivos y la precisión positiva haga `performance(predict1,"tpr","fpr")` . Guarde esto en `des` y luego extraiga los falsos positivos con `attributes(des)$x.values[[1]]*100` y la precisión positiva con `attributes(des)$y.values[[1]]*100` .
- Escriba la función que hace el gráfico de la Curva ROC y calcula el AUC. Use la siguiente función:

```
curvaROC = function(pred,y, grafico = F) {
  predict = prediction(pred,y)
  auc = attributes(performance(predict,"auc"))$y.values[[1]]*100
  des = performance(predict,"tpr","fpr")
  p = NULL
  if(grafico){
    FP = attributes(des)$x.values[[1]]*100
    PP = attributes(des)$y.values[[1]]*100
    p <- plot_ly(x = FP, y = FP, name = 'Línea No Discrimina',
                 type = 'scatter', mode = 'lines',
                 line = list(color = 'rgba(0, 0, 0, 1)',
                             width = 4, dash = 'dot'),
                 fill = 'tozeroy',  fillcolor = 'rgba(0, 0, 0, 0)') %>%
      add_trace(y = PP, name = paste('Curva ROC (AUC = ', round(auc,3),')', sep =""),
                line = list(color = 'rgba(0, 0, 255, 1)', width = 4,
                dash = 'line'),  fillcolor = 'rgba(0, 0, 255, 0.2)')%>%
      layout(title = "Curva ROC",
             xaxis = list(title = "<b>Falsos Positivos (%)<b>"),
             yaxis = list (title = "<b>Precisión Positiva (%)<b>"))
  }
  return(list(auc = auc,grafico = p))
}
```

- Haga el gráfico de la Curva ROC y calcule el AUC para los 3 modelos generados anteriormente.

7. Para calcular el KS del primer modelo obtenga el máximo de las diferencias entre precisión positiva menos los falsos positivos.

- Calcule el KS para los modelos generados en los ejercicios anteriores. Use la siguiente función:

```
KS = function(pred,y) {
  predictions = prediction(pred,y)
  des = performance(predictions,"tpr","fpr")
  ks = max(attributes(des)$y.values[[1]]*100 -
           attributes(des)$x.values[[1]]*100)
  return(ks)
}
```

- Calcule el KS para los 3 modelos generados anteriormente.
- Modifique la función `eval` para que devuelva el `AUC` y el `KS` .
- Haga una tabla con los resultados de los 3 modelos.

8. Haga 10 veces la partición de base2 en conjuntos de entrenamiento (80%) y prueba (20%) y en cada caso genere un árbol de decisión basado en el conjunto de entrenamiento, calcule los indicadores de desempeño para el conjunto de validación: e, FN, FP, AUC y KS. Haga un gráfico de estos indicadores para

ver qué tanto varían e indique la media de los mismos.

9. Haga la validación cruzada partiendo la base2 en 10 partes aproximadamente del mismo número de datos. Obtenga los indicadores de desempeño. Use la función `createFolds` de la librería `caret`. Haga un gráfico de estos indicadores para ver qué tanto varían e indique la media de los mismos.
10. Haga 10 veces la validación cruzada de base2. Haga un gráfico de las medias de los indicadores de desempeño para ver qué tanto varían.