Validacion

Problema:

Que metodos de validacion y metricas pueden predecir si un paciente ira a la sala de emergencia?

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.4.3
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
CleanData <-  read.csv("C:/Users/h703294449/OneDrive - Hofstra University/Hofstra/Service/EMALCA Honduras 20
25/Curso de Aprendizaje Estadistico/HarmonyHealthcareData.csv", stringsAsFactors = FALSE)

#names(CleanData)
#View(CleanData)

#print first 5 columns
print(colnames(CleanData)[1:5])
```

```
## [1] "Age"       "EHR.Sex"   "Ethnicity" "Language"  "Race"
```

```
#colnames(CleanData)

set.seed(42)

#To later evaluate with confusion matrix, we are portioning the orginal dataset with 80% and 20%
# p = 0.8 means 80% for training, list = FALSE returns indices
train_indices <- createDataPartition(CleanData$Admission, p = 0.80, list = FALSE)

# Create the new training dataset
new_train_data <- CleanData[train_indices, ]

# Create the new test dataset
new_test_data <- CleanData[-train_indices, ]


print("Proportions of Admission in new_train_data:")
```

```
## [1] "Proportions of Admission in new_train_data:"
```

```
print(prop.table(table(new_train_data$Admission)))
```

```
##
##       No      Yes
## 0.650864 0.349136
```

```
print("Proportions of Admission in new_test_data:")
```

```
## [1] "Proportions of Admission in new_test_data:"
```

```
print(prop.table(table(new_test_data$Admission)))
```

```
##
##        No       Yes
## 0.6507092 0.3492908
```

```
#Need to create a factor datatype (categorical) instead of character
new_train_data$Admission <- factor(new_train_data$Admission, levels = c("No", "Yes"))
new_test_data$Admission <- factor(new_test_data$Admission, levels = c("No", "Yes"))

levels(new_train_data$Admission)
```

```
## [1] "No"  "Yes"
```

```
levels(new_test_data$Admission)
```

```
## [1] "No"  "Yes"
```

Los variables idendificados son:

Before we go forward, these were the features outlined by Team 2:

Patient.HCC.Risk.Total.Risk, Active.Medications, Primary.Care.Encounter.Count, SDOH.Assessment.Count, Patient.Appointment.No.Show.Rate, Depression.Screening.Count.Past.Yr, eGFR.Result, Most.Recent.BMI.Value, UDS.Qualifying.Encounter.Result, COVID.19.Immunization.Code, Fasting.Glucose.Test.Result.

The following R Code chunk ensures that expected columns are in the csv file

```
expected_cols <- c(
  "Patient.HCC.Risk.Total.Risk", "Active.Medications", "Primary.Care.Encounter.Count",
  "SDOH.Assessment.Count", "Patient.Appointment.No.Show.Rate..", "Depression.Screening.Count.Past.Yr",
  "eGFR.Result", "Most.Recent.BMI.Value", "UDS.Qualifying.Encounter.Count",
  "COVID.19.Immunization.Code", "Fasting.Glucose.Test.Result"
)

cat("All columns exist", all(expected_cols %in% colnames(new_train_data)), "\n")
```

```
## All columns exist TRUE
```

```
# This shows us which ones don't exit if previous line was false
setdiff(expected_cols, colnames(CleanData))
```

```
## character(0)
```

```
class(new_train_data$Admission)
```

```
## [1] "factor"
```

Metodos de validacion y tecnicas

Tenemos un total de 11 variables, y para la clase que vamos a predecir es la variable `Admission` que tiene un valor binario de Yes o No, indicando si el paciente fue admitido al Departamento de Emergencia (ED)

Como nuestro metodo de validacion, vamos a utilizar K Fold Cross Validation, y para nuestras tecnicas de validacion nos enfocaremos en Recall (Especificidad y Sensibilidad) y AUC (Area Bajo la Curva ROC)

Recall es una medida de la capacidad del modelo para identificar correctamente las instancias positivas. En este caso, se refiere a la proporción de pacientes que fueron admitidos al Departamento de Emergencia (ER) y que el modelo identificó correctamente como tales.

sensitivity (también conocido como recall o tasa de verdaderos positivos) es la proporción de verdaderos positivos (pacientes admitidos correctamente identificados por el modelo) sobre el total de casos positivos reales (todos los pacientes admitidos al ER).

specificity es la proporción de verdaderos negativos (pacientes no admitidos correctamente identificados por el modelo) sobre el total de casos negativos reales (todos los pacientes no admitidos al ER).

ROC (Receiver Operating Characteristic) es una curva que muestra la relación entre la tasa de verdaderos positivos (sensibilidad) y la tasa de falsos positivos a diferentes umbrales de decisión. El área bajo la curva ROC (AUC) mide la capacidad del modelo para distinguir entre las clases positivas y negativas.

```
table(new_train_data$Admission)
```

```
##
##   No  Yes
## 1469  788
```

Como la cantidad de datos que favorece que un paciente no vaya al ER es mayor que ir al ER, vamos a utilizar la funcion smote del paquete themis para abordar nuestro problema de datos desbalanceados

De acuerdo con la documentacion, SMOTE genera nuevos ejemplos de la clase minoritaria utilizando los vecinos más cercanos de los casos minoritarios existentes. En nuestro caso, crea muestras sintéticas de la clase “Yes” al encontrar valores de predictores similares, ayudando a equilibrar el conjunto de datos y mejorar el rendimiento del modelo.

K Fold Cross Validation

```
library(themis)
```

```
## Warning: package 'themis' was built under R version 4.4.3
```

```
## Loading required package: recipes
```

```
## Warning: package 'recipes' was built under R version 4.4.3
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
##
## Attaching package: 'recipes'
```

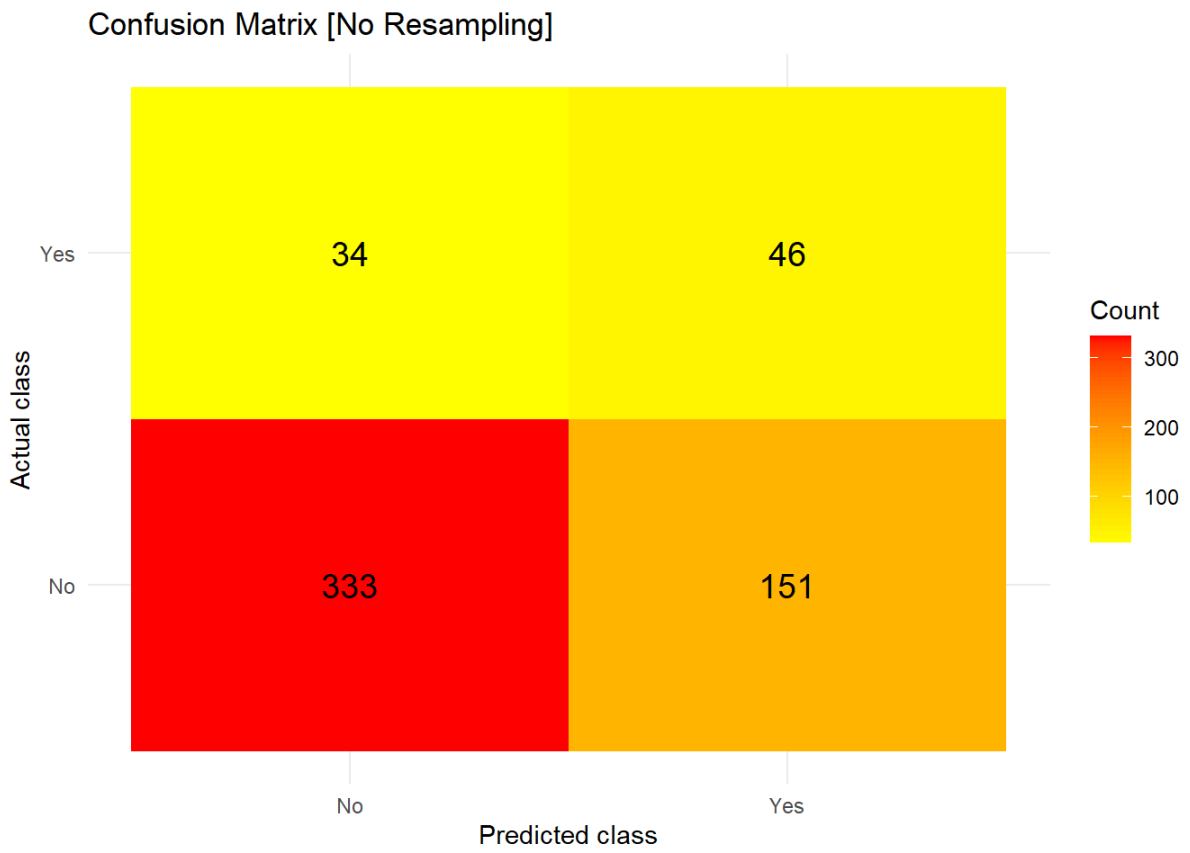
```
## The following object is masked from 'package:stats':
##
##     step
```

```
library(caret)

cv <- trainControl(
  method = "cv",
  number = 5,
  classProbs = TRUE,
  summaryFunction = twoClassSummary,
  savePredictions = "final"
)

set.seed(123)

#Important Features From Team 2
#Patient.HCC.Risk.Total.Risk+Active.Medications+Primary.Care.Encounter.Count+SDOH.Assessment.Count+Patient.A
ppointment.No.Show.Rate+Depression.Screening.Count.Past.Yr+eGFR.Result+Most.Recent.BMI.Value+UDS.Qualifying.
Encounter.Result+COVID.19.Immunization.Code+Fasting.Glucose.Test.Result

model_cv_ <- train(
  Admission ~ Patient.HCC.Risk.Total.Risk + Active.Medications +
    Primary.Care.Encounter.Count + SDOH.Assessment.Count +
    Patient.Appointment.No.Show.Rate.. + Depression.Screening.Count.Past.Yr +
    eGFR.Result + Most.Recent.BMI.Value + UDS.Qualifying.Encounter.Count +
    COVID.19.Immunization.Code + Fasting.Glucose.Test.Result,
  data = new_train_data,
  method = "glm",
  family = binomial(),
  trControl = cv,
  metric = "ROC",
  na.action = na.omit
)


print(model_cv_)
```

```
## Generalized Linear Model
##
## 2257 samples
##   11 predictor
##    2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 1805, 1805, 1806, 1807, 1805
## Resampling results:
##
##   ROC        Sens       Spec
##   0.6948276  0.8958557  0.2753608
```

Creando una matrix de confusion

```
full_pred_raw <- predict(model_cv_, newdata = new_test_data)

desired_levels <- c("No","Yes")

# Turning both to a factor
pred <- factor(full_pred_raw,    levels = desired_levels)
obs  <- factor(new_test_data$Admission, levels = desired_levels)

# create confusion matrix, with data as pred, and values to be checked as reference, with "Yes" indicating c
orrectly identifying the positive class as Went To ER
cm <- confusionMatrix(
  data      = pred,
  reference = obs,
  positive  = "Yes"
)
#Yes = WentToER, No = DidNotGoToER

cm_df <- as.data.frame(cm$table)

names(cm_df) <- c("Actual","Predicted","Count")

ggplot(cm_df, aes(x=Predicted, y=Actual, fill=Count)) +
  geom_tile() +
  geom_text(aes(label=Count), size=5) +
  scale_fill_gradient(
    low  = "yellow",
    high = "red"
  ) +
  labs(
    title = "Confusion Matrix [No Resampling]",
    x     = "Predicted class",
    y     = "Actual class",
    fill  = "Count"
  ) +
  theme_minimal()
```



Now we will be using the SMOTE resampling technqiue

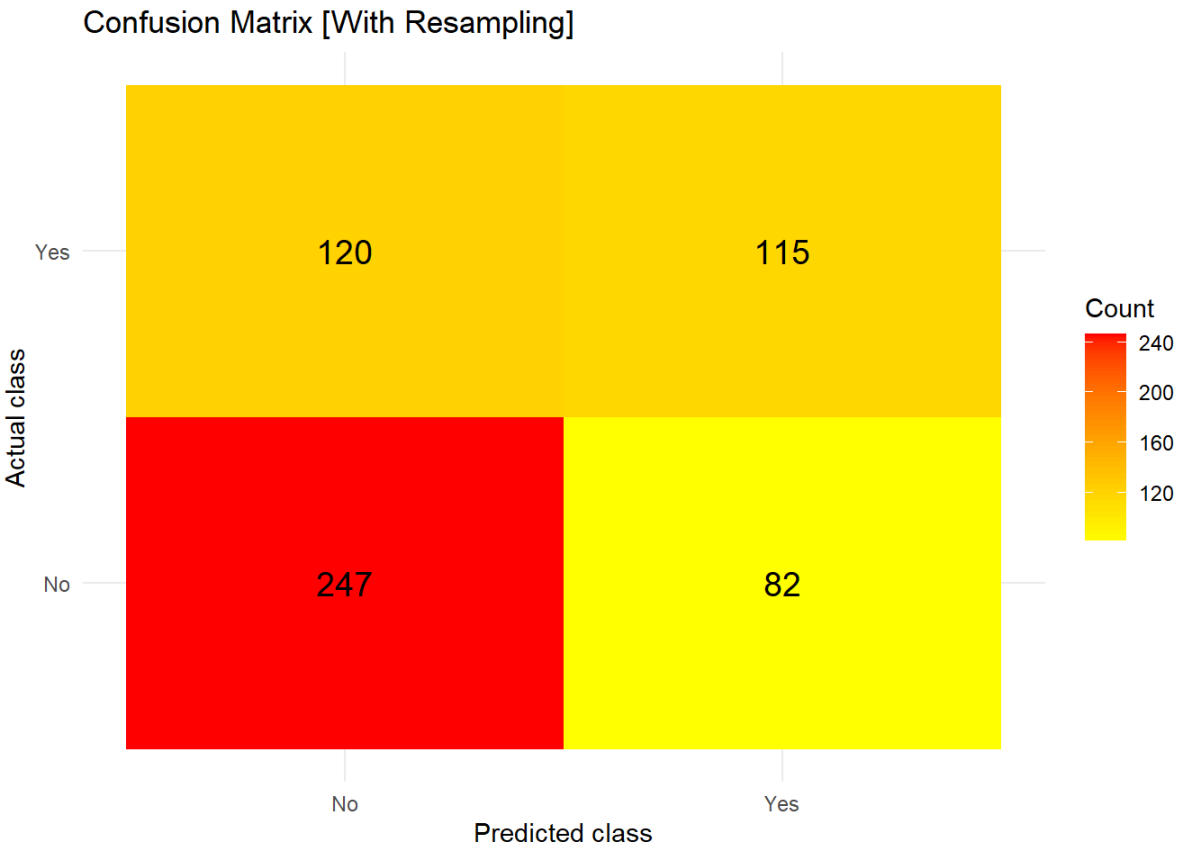
K Fold Cross Validation With Resampling

```
library(themis)
library(caret)

cv_smote_def <- trainControl(
  method = "cv",
  number = 5,
  classProbs = TRUE,
  summaryFunction = twoClassSummary,
  sampling = "smote",
  savePredictions = "final"
  )

set.seed(123)

#Important Features From Team 2
#Patient.HCC.Risk.Total.Risk+Active.Medications+Primary.Care.Encounter.Count+SDOH.Assessment.Count+Patient.A
ppointment.No.Show.Rate+Depression.Screening.Count.Past.Yr+eGFR.Result+Most.Recent.BMI.Value+UDS.Qualifying.
Encounter.Result+COVID.19.Immunization.Code+Fasting.Glucose.Test.Result

model_cv_smote <- train(
  Admission ~ Patient.HCC.Risk.Total.Risk + Active.Medications +
    Primary.Care.Encounter.Count + SDOH.Assessment.Count +
    Patient.Appointment.No.Show.Rate.. + Depression.Screening.Count.Past.Yr +
    eGFR.Result + Most.Recent.BMI.Value + UDS.Qualifying.Encounter.Count +
    COVID.19.Immunization.Code + Fasting.Glucose.Test.Result,
  data = new_train_data,
  method = "glm",
  family = binomial(),
  trControl = cv_smote_def,
  metric = "ROC",
  na.action = na.omit
)


print(model_cv_smote)
```

```
## Generalized Linear Model
##
## 2257 samples
##   11 predictor
##    2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 1805, 1805, 1806, 1807, 1805
## Addtional sampling using SMOTE
##
## Resampling results:
##
##   ROC        Sens       Spec
##   0.6944121  0.6739245  0.6154721
```

Creating Confusion Matrix for With SMOTE

```r
#using same code from earlier
#use saved model to predict on same Data set to see how it does on the dataset from Team 1
full_pred_raw <- predict(model_cv_smote, newdata = new_test_data)

desired_levels <- c("No","Yes")

# Turning both to a factor
pred <- factor(full_pred_raw,    levels = desired_levels)
obs  <- factor(new_test_data$Admission, levels = desired_levels)

# create confusion matrix, with data as pred, and values to be checked as reference, with "Yes" indicating c
orrectly identifying the positive class as Went To ER
cm <- confusionMatrix(
  data      = pred,
  reference = obs,
  positive  = "Yes"
)
#Yes = WentToER, No = DidNotGoToER

cm_df <- as.data.frame(cm$table)

names(cm_df) <- c("Actual","Predicted","Count")

ggplot(cm_df, aes(x=Predicted, y=Actual, fill=Count)) +
  geom_tile() +
  geom_text(aes(label=Count), size=5) +
  scale_fill_gradient(
    low  = "yellow",
    high = "red"
  ) +
  labs(
    title = "Confusion Matrix [With Resampling]",
    x     = "Predicted class",
    y     = "Actual class",
    fill  = "Count"
  ) +
  theme_minimal()
```


Confusion Matrix [With Resampling]
Yes
No
120
115
247
82
Actual class
No
Yes
Predicted class
Count
240
200
160
120