Herramientas Computacionales, Algoritmos y Machine Learning (HCML)

Clase 12: Modelos Regresión Logística

Constanza Prado – Alex Antequeda – Diego Muñoz

Clase 12: Modelos Regresión Logística

Regresión Logística

- Regresión Logística en R

- **Ejercicio 1:** Pacientes UCI

  - Contexto y Descripción Base de Datos.
  - Ejercicios.

- **Ejercicio 2:** Seguros

  - Contexto y Descripción Base de Datos.
  - Ejercicios.

Regresión Logística en R

Un modelo logístico en R se puede ajustar utilizando la función glm():

```
glm(formula, data, family = binomial(link = "logit"))
```

- `formula`: Fórmula indicando la relación entre la variable respuesta y las variables predictoras. La relación se indica con la virgulilla (~).

  A continuación, se presentan algunos ejemplos:
  - **Un predictor:** `respuesta ~ variable`
  - **Dos o más predictores:** `respuesta ~ predictor1 + predictor2 + ...`
  - **Todas las variables de la base como predictores:** `respuesta ~ .`

- `data`: Nombre de la base de datos que contiene las variables usadas en la definición de la fórmula.

- **family = binomial(link = "logit"):** Indica que estamos evaluando una regresión logística con función de enlace logit.

Si tenemos una variable predictora categórica catalogada en R como numérica debemos transformarla en una variable tipo `factor` para no tener problemas de codificación. En este caso, la variable respuesta puede ser tanto numérica 0-1 o `factor`.

Predicción Modelo Lineal Generalizado en R

El comando `predict.glm` permite predecir un modelo lineal generalizado en R. Como argumento, recibe un modelo glm ajustado (`object`), la base a predecir (`newdata`) y el tipo de respuesta (`type`). En el caso de una regresión logística, nos interesa extraer la predicción en forma de probabilidad, esto se define con `type="response"`:

```
probs <- predict.glm(Nombre_Modelos, newdata, type = "response")
```

Para transformar el vector de probabilidades en un vector de predicciones, se debe definir un punto de corte `c`. Una forma de realizar este vector es usando el comando `ifelse`:

```
predichos <- ifelse(probs >= c, 1, 0)
```

Matriz de confusión

La matriz de confusión (también conocida como tabla de clasificación) entrega información para evaluar la capacidad predictiva del modelo. Hay que tener en consideración lo siguiente:

1. Un modelo puede ser correcto y tener malas propiedades de clasificación.
2. En general, modelos con probabilidades estimadas cercanas a 0.5 tendrán bajo poder de clasificación.
3. Capacidad predictiva depende del punto de corte.

En las matrices de confusión se cruzan las predicciones con las tablas reales en una tabla de 2×2. En lo siguiente, positivo se refiere a los éxitos (1) y negativos se refiere a los fracasos (0):

		Valores Predichos	
		Positivos	**Negativos**
Valores Reales	**Positivos**	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Desde una matriz de confusión se pueden extraer diversos indicadores que nos permiten evaluar el modelo:

- **Sensibilidad:** = $\frac{VP}{VP+FN}$
- **Precisión** = $\frac{VP}{VP+FP}$
- **Especificidad:** = $\frac{VN}{VN+FP}$
- **Exactitud** = $\frac{VP+VN}{VP+VN+FP+FN}$

La **sensibilidad** representa la proporción de positivos capturados correctamente por el modelo, sobre el total de positivos reales. Representa qué tan bien el modelo califica los casos "positivos" de nuestros datos.

La **especificidad** representa la proporción de casos negativos capturados correctamente por el modelo, sobre el total de negativos reales. Representa qué tan bien el modelo califica los casos "negativos" de nuestros datos.

La **exactitud** mide la proporción de casos clasificados correctamente, independiente de si es positivo o negativo. Esta medida no es recomendada cuando la base de datos es desbalanceada.

La **precisión** es el porcentaje de casos positivos clasificados correctamente.

Matriz de confusión en R

La función `confusionMatrix` de la librería `InformationValue` ajusta la matriz de confusión entre valores reales y valores predichos. Esta función recibe los valores reales de los datos (`actuals`), la probabilidad ajustada por el modelo (`predictesScores`) y el punto de corte (`threshold`):

```
InformationValue::confusionMatrix(actuals, predictedScores, threshold)
```

También puede usarse la función `ConfusionMatrix` de la librería `MLmetrics`. En este caso, los argumentos son los valores reales de los datos `y_true` y el vector de predicciones `y_pred`:

```
MLmetrics::ConfusionMatrix(y_pred, y_true)
```

Indicadores de ajuste en R

En la librería `MLmetrics` existen diferentes funciones que permiten calcular algunos de los indicadores anteriormente mencionados. A continuación, presentamos las funciones para sensibilidad, especificidad y exactitud, respectivamente:

```
MLmetrics::Sensitivity(y_true, y_pred, positive = '1')
MLmetrics::Specificity(y_true, y_pred, positive = '1')
MLmetrics::Accuracy(y_true, y_pred)
```

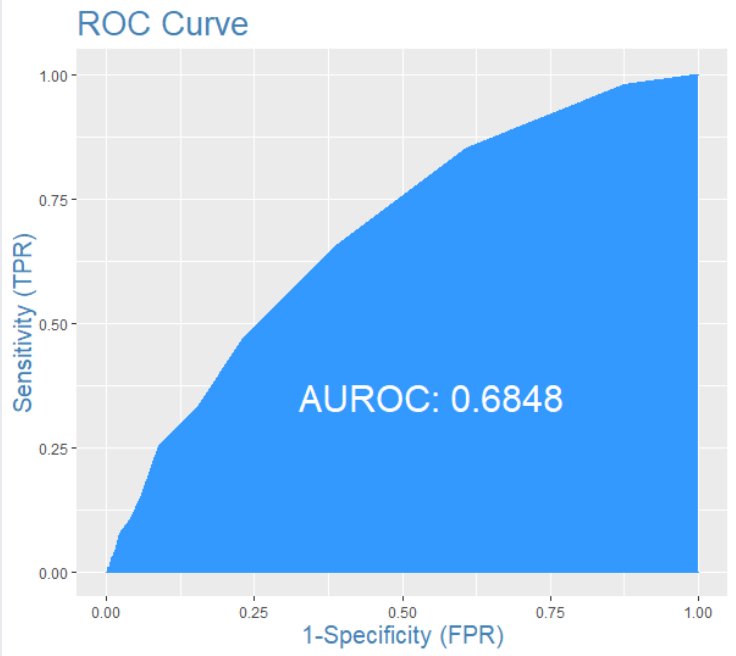
Todos reciben como argumento los valores reales de los datos `y_true` y el vector de predicciones `y_pred`. Además, existe el argumento `positive`, el cual nos permite seleccionar cual es el factor catalogado como positivo en nuestros datos. En este caso, se usa `positive = '1'`

La librería `InformationValue` contiene funciones que computan la sensibilidad (`sensibility`) y la especificidad (`specificity`). Estas funciones tienen como argumento los valores reales (`actuals`), las probabilidades ajustadas (`predictedScores`) y el punto de corte (`threshold`):

```
InformationValue::sensitivity(actuals, predictedScores, threshold)
InformationValue::specificity(actuals, predictedScores, threshold)
```

Curva ROC

Una curva ROC representa la tasa de verdaderos positivos (sensibilidad) frente a la tasa de falsos positivos (1-especificidad) en diferentes umbrales de clasificación. Reducir el umbral de clasificación clasifica más elementos como positivos, por lo que aumentarán tanto los falsos positivos como los verdaderos positivos.



Desde la curva ROC se obtiene el indicador AUC (área bajo la curva). Este representa el área bajo la curva ROC, entregando valores entre 0.5 (50%) y 1 (100%). Una forma de interpretar el AUC es la probabilidad que el modelo asigne una probabilidad más alta a un caso positivo que un caso negativo.

La función `plotROC` de la librería `InformationValue` ajusta la curva ROC del modelo. Esta función recibe los valores reales (`actuals`) y las probabilidades ajustadas (`predictedScores`).

Ejercicio 1: Muertes UCI

Mortalidad de pacientes en Unidad de Cuidados Intensivos (UCI)

El archivo `UCI.csv` contiene registros de pacientes ingresados a Unidad de Cuidados Intensivos. El objetivo es estudiar el Estado Vital de los pacientes, el cual puede tomar dos valores: Vivo (0) o fallecido (1). Para esto, usted sabe que la regresión logística modela la probabilidad de éxito (en este caso, fallecimiento del paciente) permitiendo comprender qué variables pueden incidir en una mayor mortalidad.

Las variables de la base de datos son:

Variable	Descripción
ID	Código único de paciente
STA	Estado vital (1: fallecido, 0: vivo)
AGE	Edad en años
SEX	Sexo (1: mujer, 0: hombre)
RAC	Raza (1: blanco, 2: negro, 3: otra)
CAN	Presencia de cáncer (2: sí, 1: no)
CRN	Problemas al riñón (2: sí, 1: no)
INF	Infección al ingreso (2: sí, 1: no)
CPR	Necesidad de resucitación pulmonar (2: sí, 1: no)
SYS	Presión sanguínea sistólica al ingreso (mm Hg)
HRA	Frecuencia cardíaca (latidos por minuto)
PRE	El paciente fue admitido previamente en la UCI hace 6 meses (2: sí, 1:no)
TYP	Tipo de admisión (2: emergencia, 1: electiva)

Paquetes a utilizar:

```
library(readr)
library(dplyr)
library(ggplot2)
```

---

a) Revise cuál es el formato de las variables dicotómicas presentes en la base de datos. ¿Por qué pudiera ser importante notar su formato al utilizar modelos de regresión? Comente.

b) Explique por qué no tiene sentido aplicar un modelo de regresión lineal cuando la variable respuesta es una variable dicotómica.

c) Escriba la ecuación del modelo de regresión logística de STA sobre SYS. Interprete qué es lo que realiza una regresión logística.

d) Plantee el modelo anterior en R y realice análisis de significancia con un 90% de confianza de la variable Presión sistólica. Comente.

e) En clases vimos cómo podemos ir seleccionando variables 1 a 1. Por ejemplo, con el método forward (se añaden variables hasta que el ingreso de más variables no aporte al modelo). Utilice este método hasta que no hayan más variables significativas para agregar. ¿Qué variables entran en el modelo? ¿Cuáles quedan fuera?

Hint: La función `add1()` indica la significancia de las variables para priorizar cuál añadir.

f) Interprete los coeficientes del modelo obtenido anteriormente. ¿Qué variables incrementan la probabilidad de fallecer al ingresar a UCI?

Ejercicio 2: Seguros

Tenencia de un seguro complementario

El archivo *seguros.csv* contiene características de una cartera de clientes y lo que interesa conocer es cuáles de estas características pueden asociarse con una tenencia de seguro complementario.

Las variables de la base de datos son:

Variable	Descripción
Seguro	Indica si cuenta con seguro complementario (1: sí, 0: no)
Edad	Edad en años
Chile	Indica si la persona vivió fuera de Chile por más de 10 años (1: sí, 0: no)
Sexo	Sexo (1: mujer, 0: hombre)
Educ	Años de educación
EstCiv	Estado Civil (1: casado, 0: soltero)
Salud	Estado de salud (1: Excelente, 2: Muy buena, 3: Buena, 4: Regular, 5: Mala)
Jubil	Situación laboral (2: trabajando, 1: jubilado)
Ingreso	Ingreso per Cápita
IngresoQuintil	Categorización en quintiles de ingreso

a) Revise cuál es el formato de las variables dicotómicas presentes en la base de datos. Genere una base de entrenamiento con el 70% de los datos y una base de testeo con el 30% restante.

b) Con la base de entrenamiento, ajuste un modelo de regresión logística para la variable respuesta Seguro, ocupando la variable Jubil como predictora. Interprete la chance obtenida para esta variable.

c) En clases vimos cómo podemos ir seleccionando variables según AIC. Por ejemplo, con método forward se incluyen variables hasta obtener el mejor AIC. Utilice este método hasta que no haya más variables para agregar. ¿Qué variables entran en el modelo? ¿Cuáles quedan fuera?

Hint: La función `step()` selecciona el mejor modelo según AIC.

d) Interprete los coeficientes del modelo obtenido anteriormente. ¿Qué variables incrementan la probabilidad de tener un seguro complementario?

e) Con la función `optimalCutoff()` del paquete `InformationValue`, determine cuál es el punto de corte óptimo para predecir si un cliente tiene seguro complementario o no.

Nota: Con la función `optimalCutoff()` por defecto elige el punto de corte que minimiza el error de clasificación.

f) Genere las predicciones para la base de testeo ocupando el modelo obtenido bajo el criterio de AIC.

g) Evalúe la calidad de predicción del modelo con el punto de corte encontrado, calculando los siguientes indicadores:

- Curva ROC
- Sensibilidad
- Especificidad
- Precisión

Reporte los valores para la base de entrenamiento como para la de testeo. Comente las diferencias encontradas

¡Gracias!