TRABAJO FIN DE MÁSTER

VALIDACIÓN INTERNA DE MODELOS PREDICTIVOS DE REGRESIÓN LOGÍSTICA. COMANDO VALIDATION (STATA)

Septiembre 2018

Máster en Bioestadística

Facultad de Estudios Estadísticos

Universidad Complutense de Madrid

BORJA MANUEL FERNÁNDEZ FÉLIX

TUTORES:
TERESA PÉREZ PÉREZ Y ALFONSO MURIEL GARCÍA

Índice

Resumen

El desarrollo de modelos predictivos de regresión logística es uno de los métodos estadísticos más empleados en el área de la medicina. La probabilidad estimada a partir de estos modelos se emplea en dos contextos distintos: pronóstico, en el cual se desea determinar la probabilidad de que un evento específico ocurrirá en el futuro; y diagnóstico, donde el objetivo se centra en la probabilidad de que cierta condición o enfermedad esté presente. Pero antes de que un modelo predictivo sea implementado en la práctica clínica debe ser validado interna y externamente.

La declaración TRIPOD, recomendaciones basadas en la evidencia para el reporte de estudios de modelos predictivos, sugiere que las técnicas de validación interna tienen que ser reportadas. A pesar de ello, varias revisiones sistemáticas han demostrado que la validación interna y externa de los modelos predictivos es poco frecuente.

Con el objetivo de aumentar la frecuencia de validación en los modelos predictivos en este trabajo se exponen las diferentes técnicas de validación interna: aparente, por división de datos, cruzada y bootstrap; y se ha desarrollado una herramienta sencilla, el comando validation, en el programa Stata para realizar la validación interna de un modelo predictivo mediante técnicas bootstrap.

El comando validation permite evaluar los dos aspectos principales del rendimiento de un modelo predictivo: la discriminación y la calibración. La capacidad discriminante del modelo es evaluada mediante el C-Statistic aparente y ajustado por el optimismo. El rendimiento en términos de calibración se reporta mediante el test de Hosmer-Lemeshow y la pendiente de calibración o Shrinkage factor. El comando incluye una opción gráfica que permite obtener la curva ROC, los histogramas de probabilidades predichas por el modelo en función del evento de interés y un gráfico de calibración con las probabilidades predichas vs. las observadas separadas por grupos de riesgo. Como complemento el comando reporta el número de veces que cada predictor es incluido en el modelo final en las muestras bootstrap.

Este comando permitirá a los investigadores realizar de una forma sencilla la validación interna de los modelos predictivos de regresión logística empleando técnicas bootstrap.

1. Introducción

1.1. Predicción

Anunciar por revelación, conocimiento fundado, intuición o conjetura algo que ha de suceder. Este es el significado de predecir según la Real Academia Española – RAE.

En medicina, la mayoría de decisiones de los profesionales se toman basadas en la probabilidad o riesgo de un paciente. La probabilidad se emplea en dos contextos o entornos distintos: entorno pronóstico, en el cual se desea determinar la probabilidad de que un evento específico ocurrirá en el futuro; y entorno diagnóstico, donde el objetivo se centra en la probabilidad de que cierta condición o enfermedad esté presente.

Pronosticar, en general, significa prever, predecir o estimar la probabilidad o riesgo de una condición futura. Todos estamos habituados a escuchar a diario pronósticos sobre el tiempo o la economía. En medicina, el pronóstico habitualmente lo asociamos a la probabilidad o riesgo de que un individuo desarrolle un particular estado de salud en un tiempo determinado basado en su perfil clínico y no clínico. El interés suele centrarse en eventos tales como muerte, complicación de la enfermedad, curación, reingreso o cambios en el dolor o la calidad de vida. En el entorno pronóstico las predicciones pueden ser usadas para planificar decisiones terapéuticas o estilos de vida en base al riesgo de desarrollar un especifico estado de salud. Ampliamente conocido y usado es el score de riesgo cardiovascular de Framingham. [1]

En la Antigua Grecia, Hipócrates, una de las figuras más destacadas en la historia de la medicina, ya puso en valor el estudio pronóstico de las enfermedades. La importancia asociada al pronóstico era una de las características principales del sistema Hipocrático de medicina, gracias al cual se tenía un conocimiento completo del estado anterior y actual del paciente y la tendencia propia de la enfermedad.

"The physician who cannot inform his what would be the probable issue of his complaint if to follow its natural course is not qualified to prescribe rational plan of treatment for its cure." [2]

Traductor de Hipócrates

Diagnosticar, en general, significa examinar una cosa, un hecho o una situación para realizar un análisis o para buscar una solución a sus problemas o dificultades. En medicina, identificar una enfermedad mediante un examen de los signos y los síntomas que presenta cierto individuo o paciente. En el entorno diagnóstico, las predicciones pueden ser usadas en la toma decisiones tales como llevar a cabo nuevas pruebas más específicas, quizás más invasivas para el paciente o más costosas, para el inicio temprano de un tratamiento, o simplemente, para tranquilizar al paciente porque su probabilidad asociada a la presencia de cierta enfermedad es muy baja. Un ejemplo diagnóstico muy conocido es el uso de un test prenatal para predecir el riesgo de una mujer embarazada de tener un bebé con síndrome de Down.

1.2. Modelos predictivos

Dada la variabilidad existente entre pacientes y en la presentación, el tratamiento y la etiología de una enfermedad y otros estados de salud, la información de una única variable o predictor raramente será suficiente para dar una estimación adecuada de la predicción. En la toma de decisiones de los médicos se utiliza, de forma implícita o explícita, la probabilidad o riesgo del paciente. Esta probabilidad habitualmente se basa en la combinación de múltiples predictores de un individuo que han sido observados y medidos. Por tal motivo, es necesario el uso de métodos multivariables tanto en el diseño como en el análisis de los estudios predictivos. De modo que podamos determinar los predictores, o las diferentes combinaciones de estos, que permitan estimar la probabilidad de la forma más ajustada posible. Estas herramientas son comúnmente llamadas modelos predictivos, modelos pronóstico, reglas de predicción o puntuaciones o scores de riesgo.

Un modelo de predicción multivariable es una ecuación matemática que relaciona múltiples predictores de un particular individuo con la probabilidad de presencia – entorno diagnóstico – u ocurrencia futura – entorno pronóstico – de un resultado.

Los modelos predictivos son usados en diferentes contextos, desde niveles de atención primaria hasta los niveles más especializados. En la mayoría de los ámbitos médicos, se están desarrollando, validando, actualizando y, en algunos casos, implementando modelos predictivos con el objetivo de ayudar a los médicos e individuos en la toma de decisiones, tales como tratar o no tratar, derivar a una asistencia más especializada, etc.

La principal diferencia entre los modelos predictivos con objetivo pronóstico o diagnóstico es el tiempo para el cual se estiman las probabilidades. Mientras que en el entorno pronóstico las probabilidades se basan en eventos que pueden suceder en un futuro, en el entorno diagnóstico la probabilidad recae sobre el estado actual del paciente. Pero a pesar de los diferentes *timing* de la predicción, existen muchas similitudes entre los modelos de predicción pronósticos y diagnósticos:

- La variable de resultado – outcome – suele ser binaria, ya sea presencia o ausencia de cierta patología (en diagnóstico) u ocurrencia o no de cierto evento (en pronóstico).
- El interés principal se centra en estimar la probabilidad del outcome a partir de dos o más variables predictoras.
- Los pasos a seguir en el desarrollo del modelo, tales como la selección de predictores, las estrategias de construcción del modelo, el manejo de variables continuas y el peligro de sobreajuste son los mismos.
- Las medidas para evaluar el rendimiento del modelo son las mismas.

En medicina se han desarrollado y se siguen desarrollando muchos modelos predictivos, pero luego muchos de estos no son implementados en la práctica clínica. Esto se debe a varios motivos. Primero, muchos de los modelos que se desarrollan son demasiado complejos para implementar en la práctica diaria de un clínico, aunque actualmente los ordenadores ayudan a resolver esta dificultad, y es frecuente entre los investigadores desarrollar calculadoras online que permitan estimar las probabilidades predichas por el modelo rápidamente. Segundo, porque los clínicos en muchas ocasiones no saben cómo utilizar las probabilidades o riesgos obtenidos con el modelo en la toma de decisiones. Y finalmente, debido a que muchos modelos predictivos no han sido validados, lo cual genera desconfianza en los clínicos.

Este último motivo, la validación de los modelos, concretamente la validación interna que se explicará más adelante, será el objeto de estudio de este trabajo.

1.3. Validación de los modelos predictivos

Una vez se ha encontrado el mejor modelo posible, hay que validarlo, es decir, comprobar si el modelo creado funciona igual en otros individuos distintos a los que se han empleado en el desarrollo del modelo. [3]

En los modelos predictivos validar significa ver si el modelo predice bien la variable dependiente en nuevos individuos. Existen dos modos de validación, externa e interna.

- La **validación externa** es la validación más estricta, consiste en emplear el modelo predictivo desarrollado en otra muestra de pacientes, a poder ser de otra área geográfica y/o en otro periodo de tiempo.
- La **validación interna** es menos estricta pues usa los datos de los sujetos a partir de los cuales se ha desarrollado el modelo predictivo, pero no necesita del esfuerzo añadido de reclutar nuevos individuos y, por tanto, está al alcance del propio investigador en el momento que desarrolla el modelo.

Antes de recomendar el uso en la práctica clínica de un modelo predictivo los autores solicitan que se haya examinada la estabilidad, la reproductividad y la validez externa del modelo en una muestra de datos independientes de aquella utilizada en el desarrollo del mismo. [4] No obstante, la validación interna debe evaluarse en todos los modelos predictivos. De este modo muchas validaciones externas fallidas podrían ser resueltas previamente gracias a una rigurosa validación interna, con el ahorro que supone en tiempo y recursos invertidos. Una revisión reciente ha constatado que los estudios de desarrollo de modelos predictivos a menudo son realizados a partir de tamaños de muestra relativamente pequeños para la complejidad empleada en los procesos de selección de variables y estimación de los efectos. La mediana del tamaño de muestra de estos estudios fue de 445 sujetos. [5] El factor limitante en este tipo de investigación es el número de eventos, o no eventos (el menos frecuente), y muchos de los modelos desarrollados están lejos de las recomendaciones del ratio de número de eventos por predictores evaluados. En estos casos de tamaño de muestra pequeño la validación interna es esencial.

La declaración TRIPOD [6][7] (Transparent Reporting of multivariable prediction model for Individual Prognosis or Diagnosis) es un conjunto de recomendaciones basadas en la evidencia para informar de forma estándar los estudios de modelos de predicción en ciencias biomédicas. Incluye 22 ítems que deben ser informados, con el objetivo de

mejorar el reporte transparente en los estudios de desarrollo, validación o actualización de modelos de predicción, sean con fines diagnósticos o pronósticos.

Entre la lista de ítems que se deben informar en un estudio de desarrollo de un modelo predictivo multivariante se incluye, en el apartado de métodos de análisis estadísticos, que se especifique el método de validación interna del modelo empleado.

Sin embargo, uno de los principales déficit en los estudios de modelos predictivos se centra en la validación de los modelos. Varias revisiones sistemáticas en diferentes ámbitos de la medicina han mostrado que apenas un 10% de los modelos desarrollados son validados externamente. [8][9] Y a pesar de que para realizar la validación interna del modelo no es necesario un esfuerzo añadido en la recogida de más datos, tan sólo, entre el 30% y el 40% de los modelos de la literatura son validados internamente. [8][9]

1.4. Software estadístico

Uno de los motivos de la ausencia de validación de los modelos predictivos se puede achacar a la carencia de herramientas al alcance de los investigadores que permitan validar de forma sencilla los modelos. En ciencias biomédicas el uso y manejo de software estadístico es habitual entre los científicos, y dos de los programas más utilizados por la comunidad son Stata [10] y R. [11]

Ante este déficit de herramientas los usuarios no se han quedado parados, tanto para Stata como para R ya se pueden encontrar diferentes comandos o macros que dan acceso a los investigadores a diferentes métodos de validación interna de modelos de predicción.

En el software Stata, cuya versión más actual es Stata 15.1, se puede encontrar algunos comandos, tales como *crossfold*, *calibrationbelt*, *bootstrap* o *swboot* que nos permiten realizar distintos tipos de validación interna. Así como en el software R, la versión más actual es la versión 3.5.1, con paquetes como *calibrate*, *plot.calibration* o *validate*. Más adelante, cuando se expliquen los tipos de validación interna, serán expuestos con detalle.

Sin embargo, y pese a tener disponibles varias herramientas que permiten realizar algún tipo de validación interna, completa o parcialmente, del modelo, estos no son utilizados o, al menos, no son informados con la frecuencia deseada. Hay varios motivos. Primero, la falta de conocimiento de los diferentes métodos de validación interna que existen y

que se exponen en este trabajo. Segundo, que los investigadores que están desarrollando los modelos, a pesar de ser usuarios "habituales" de estos softwares, desconocen los comandos o herramientas ya generadas que permiten validar internamente el modelo de forma sencilla. Tercero, que ninguna de estas herramientas disponibles aporta la información completa que se necesita para validar un modelo, y la información no se presenta de una forma atractiva para el investigador. Y cuarto, que en algunas ocasiones se necesitan potentes ordenadores.

2. Objetivo

El objetivo principal del trabajo es exponer los diferentes métodos de validación interna y desarrollar un comando en el software estadístico Stata que permita realizar la validación interna de un modelo predictivo empleando técnicas de remuestreo bootstrap de forma sencilla y atractiva para los investigadores.

2.1. Descripción de los diferentes métodos de validación interna

Como se exponía anteriormente la validación interna es menos estricta, pues usa los datos de los sujetos a partir de los cuales se ha desarrollado el modelo predictivo. Sin embargo, la ventaja que tiene es que no necesita del esfuerzo añadido de reclutar nuevos individuos y, por tanto, está alcance del propio investigador en el momento que desarrolla el modelo.

Existen diferentes técnicas de validación interna: [7]

- **Apparent validation** o rendimiento aparente. Consiste en evaluar el rendimiento del modelo utilizando los mismos datos empleados en el desarrollo del mismo.
- **Data splitting** o división de los datos. El método más simple de validación consiste en dividir de forma aleatoria la muestra original en dos submuestras, una donde se desarrolla el modelo – training – y otra donde se valida – test –.
- **Cross-validation** o validación cruzada. Una extensión de las técnicas de splitting que reduce el sesgo y la variabilidad de la estimación del rendimiento.
- **Bootstrap validation.** Este método, como la validación cruzada, utiliza todos los datos empleados en el desarrollo del modelo. Además, nos permite cuantificar el optimismo del modelo de predicción final.

2.2. Comando validation

Ante la carencia de herramientas de manejo sencillo en los softwares más habituales entre los investigadores en ciencias biomédicas, el objetivo del estudio se ha centrado en crear un comando – *validation* – en el software Stata que permita, en el momento de desarrollo del modelo, llevar a cabo la validación interna. Entre los distintos métodos de validación interna las técnicas de remuestreo permiten obtener estimaciones más

realistas del rendimiento del modelo [12]. Por ello, el comando desarrollado está basado en técnicas de remuestreo bootstrapping.

3. Metodología

El rendimiento predictivo de un modelo evaluado en los mismos datos que han sido empleados para generar el modelo se conoce como el rendimiento aparente del modelo. Muchos modelos de predicción son sobreajustados y su rendimiento aparente es optimista. Esto en muchos casos es debido a la inclusión de un gran número de predictores en relación al número de eventos de la variable de resultado. La recomendación más habitual es incluir un predictor por cada 10 eventos – o no eventos – el menos frecuente. [13] Otra causa habitual de optimismo es el uso de estrategias de selección de predictores, que se acentúa cuando se trabaja con muestras de tamaño pequeño [12][14]. Cuando el tamaño de muestra de la base de desarrollo del modelo es realmente grande el optimismo en la estimación del rendimiento aparente disminuye. [15]

Un modelo ajustado con n-variables para n+1 observaciones predecirá perfectamente la variable Y, salvo que varias observaciones tengan la misma Y. Pero este modelo cuando se utilice en otro conjunto de datos nos dará estimaciones que parecerían haberse creado al azar. Este es el caso más extremo de sobreajuste pero indica la importancia que tiene que las estimaciones del rendimiento predictivo del modelo sean insesgadas. Por ello, es recomendado que los estudios de desarrollo de un modelo predictivo incluyan algún método de validación interna para obtener una estimación más realista del rendimiento del modelo.

No obstante, después de desarrollar un modelo predictivo y validarlo internamente, es altamente recomendable evaluar el rendimiento del modelo en una muestra externa.

Los predictores candidatos se pueden obtener a partir de las características demográficas de los pacientes, la historia clínica, examinación física, características de la enfermedad, resultados de test y tratamientos previos. Estos deben ser claramente definidos, estandarizados y reproducibles para que puedan ser generalizados y de utilidad en la práctica clínica.

Existen diferentes estrategias de selección de potenciales predictores. Una estrategia ampliamente utilizada, pero que en la práctica puede reportar modelos sesgados, es la selección de predictores a partir de la asociación estadística con el outcome de interés

mediante un análisis univariante. La selección de predictores mediante esta estrategia puede dar lugar a modelos en los que variables importantes se queden fuera de modelo máximo planteado por no existir asociación estadísticamente significativa, quizás porque no había suficiente potencia debido al escaso tamaño muestral para contrastar dicha asociación. En el lado opuesto, si el tamaño muestral es muy grande cualquier diferencia, por pequeña que sea, será estadísticamente significativa aunque no tenga relevancia clínica, y en el modelo máximo podríamos tener problemas de sobreajuste por incluir demasiadas variables. Otra estrategia de selección de variables predictoras es aquella basada únicamente en la experiencia previa y/o conocimiento a través de la literatura del investigador. Es decir, sin entrar en los datos, el investigador decide las variables que van a ser incluidas en el análisis. La tercera estrategia consiste en una mixtura entre las dos anteriores, el conocimiento del investigador y la información que aportan los datos.

Los modelos de predicción que nos encontramos más habitualmente en el ámbito sanitario son modelos de regresión logística. Estos modelos se emplean cuando la variable dependiente – el outcome – es binaria. La forma de un modelo de regresión logística es la siguiente:

$$\ln\left({}^{p}/_{q}\right) = \alpha_0 + \alpha_1 x_1 + \cdots + \alpha_k x_k$$

Siendo ln el logaritmo neperiano, p la probabilidad de éxito y q su complementario, α_0 la constante del modelo, α_1 ... α_k los coeficientes de los predictores incluidos en el modelo y x las variables predictoras del modelo.

Alternativamente,

$$p = \frac{1}{1 + e^{-(\alpha_0 + \alpha_1 x_1 + \cdots + \alpha_k x_k)}}$$

Esta última expresión, si son conocidos los coeficientes, permite calcular directamente la probabilidad del proceso binomial para los distintos valores de la variable x.

El outcome de un estudio predictivo debe ser relevante no sólo para el investigador sino también para el paciente. Outcome tales como la ocurrencia o remisión de la enfermedad, la muerte, el dolor o la calidad de vida son habituales en investigación biomédica.

El diseño ideal para responder a una cuestión predictiva es un estudio de cohortes, preferiblemente prospectivo, aunque es frecuente encontrar estudios retrospectivos en la literatura.

3.1. Medidas de rendimiento de los modelos predictivos

Validar un modelo predictivo desarrollado implica evaluar su rendimiento. Para evaluar el rendimiento de un modelo de regresión logística, concretamente, son dos los aspectos principales que se deben valorar: la discriminación y la calibración.

3.1.1. Discriminación

Se refiere a la capacidad del modelo para distinguir entre los individuos quienes experimentan el evento de interés y los individuos que no experimentan el evento de interés. Un modelo predictivo discrimina perfectamente cuando la probabilidad predicha para todos los individuos que tienen el evento de interés es mayor que la de todos los individuos que no lo tienen (figura 2).

La capacidad discriminante de un modelo de regresión logística frecuentemente se evalúa usando el estadístico de concordancia C-Statistic. Este se puede calcular tomando todos los posibles pares de sujetos donde uno experimenta el evento y el otro no. El C-Statistic es la proporción de todos los pares en los cuales la probabilidad predicha para el individuo que experimentó el evento es mayor que la predicha para el individuo que no experimentó el evento. [16]

La discriminación de un modelo de regresión logística también puede ser descrita mediante el área bajo la curva ROC (Receiver Operating Characteristic), a menudo denotada por AUC. Refleja la probabilidad de que seleccionados al azar un par de individuos, uno con el evento y otro sin el evento, el modelo asigne la probabilidad más alta al individuo con el evento. En regresión logística el estadístico de concordancia y el área bajo la curva ROC son equivalentes. [17]

El área bajo la curva ROC se suele presentar también de forma gráfica, presentando la sensibilidad en función de los falsos positivos (complementario de la especificidad) para distintos puntos de corte (Figura 1). El AUC oscila entre 1 – discriminación perfecta – y el 0.5, que indica que el modelo discrimina igual que si tirásemos una moneda al aire para decidir si el individuo tendrá el evento o no. Un área bajo la curva ROC superior a

0.7 la discriminación del modelo se considera aceptable. [18] Otros autores consideran que el modelo tendrá cierta utilidad predictiva si supera 0.8. [16]

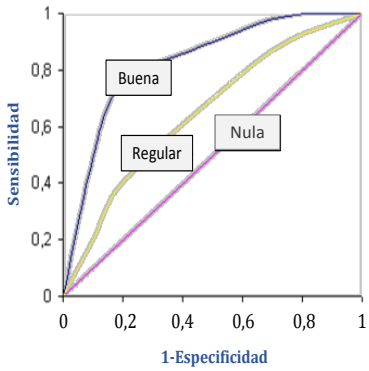

Figura 1. Distintos tipos de curvas ROC

Existen otros estadísticos que suelen emplearse para cuantificar la capacidad discriminante de un modelo logístico como el estadístico D_{xy} de Somers.[19] Se puede calcular a partir de la siguiente fórmula:

$$D_{xy} = 2 \times (AUC - 0.5)$$

Su interpretación es similar a la del área bajo la curva ROC, excepto que este se distribuye en una escala de -1 a 1. Cuanto más próximo está de la unidad mejor discrimina el modelo. Estos índices basados en rangos tienen la ventaja de ser independientes a la prevalencia del evento.

Gráficamente también existen otros métodos para evaluar la discriminación del modelo. Además de la curva ROC, la discriminación del modelo se puede evaluar mediante los histogramas de probabilidades de los individuos que desarrollan el evento versus los que no desarrollan el evento (figura 2). Estas mismas probabilidades pueden presentarse mediante los gráficos de cajas y bigotes.

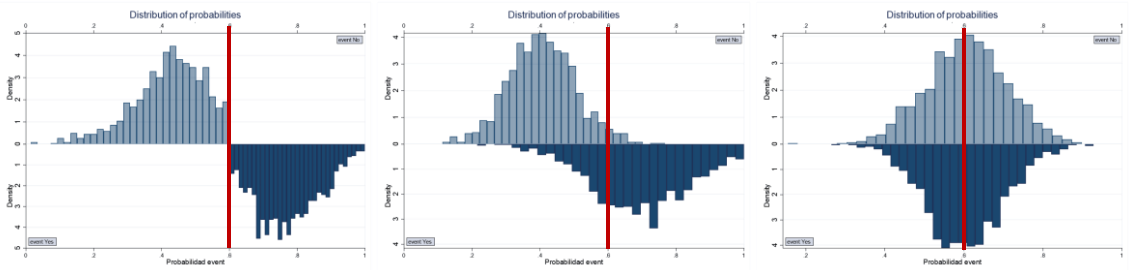

Figura 2. Distribuciones de probabilidad por evento para diferentes capacidades discriminantes

3.1.2. Calibración

Refleja el grado de acuerdo entre las predicciones estimadas por el modelo y los resultados observados.

La prueba estadística que evalúa la calibración del modelo es el test de Hosmer-Lemeshow Goodness of Fit (GOF). La idea del test es la siguiente, si el ajuste del modelo es bueno, un valor alto de la probabilidad predicha se asociará con el resultado 1 de la variable dependiente. Se trata de calcular para cada individuo del conjunto de datos la probabilidad del evento pronosticada por el modelo, agruparlas en grupos de riesgo y calcular, a partir de ellas, la frecuencia de eventos esperados en cada grupo. Basada en el estadístico Chi-cuadrado se contrasta la hipótesis de que la frecuencia de eventos predichos y observados en cada grupo de riesgo son iguales. Tiene el inconveniente, como todos los test frecuentistas, que ante tamaños muestrales grandes por pequeñas diferencias que aparezcan estas resultan estadísticamente significativas. Para el número de grupos de riesgo habitualmente se emplean deciles o cuartiles.

Un método muy utilizado para evaluar la calibración es mediante un gráfico de calibración que confronta las probabilidades predichas por el modelo (en el eje x) versus las probabilidades observadas (en el eje y). Habitualmente se realiza por deciles de riesgo. Con este gráfico podemos ver la dirección y magnitud de la mala calibración, si la hubiera, a través del rango de probabilidades.

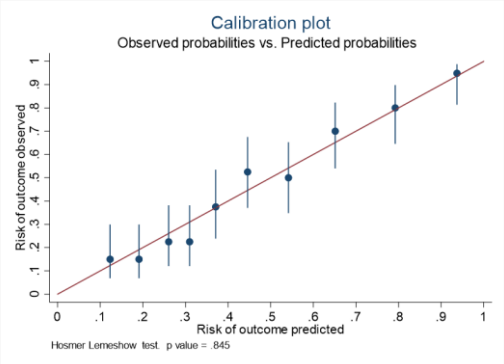

Figura 3. Gráfico de calibración

Un último método para evaluar la calibración del modelo predictivo es la pendiente de calibración – calibration slope –. El predictor lineal viene definido como la suma en la escala logarítmica de los coeficientes de regresión (estimados en la muestra de desarrollo) multiplicado por el valor del individuo en el predictor correspondiente. La

pendiente por definición es uno en la muestra de desarrollo. En un conjunto de validación o en las muestras bootstrap, cuando aplicamos esta técnica, la pendiente de calibración puede ser estimada usando un modelo de regresión logística que incluya como única variable independiente el predictor lineal. Una calibración perfecta mostraría una pendiente de 1, pero generalmente, debido al sobreajuste la pendiente de calibración es inferior. Esto indica que los coeficientes que fueron estimados en la muestra de desarrollo deben ser estrechados. La pendiente de calibración también es conocida como Shrinkage factor, o factor de contracción. El Shrinkage factor es un método de estimación estadística que permite acercar los coeficientes de regresión del modelo hacía el cero, de modo que la calibración del modelo en un nuevo conjunto de datos sea mejor. [17] Van Houwelingen y le Cessie proporcionaron un estimación heurística del shrinkage que demostraron que funcionaba bien en diferentes contextos. [21]

$$\widehat{\gamma} = \frac{model\ X^2 - p}{model\ X^2}$$

Donde p es el número total de grados de libertad para los predictores y $model\ X^2$ es el estadístico X^2 de la razón de verosimilitud para testar la influencia conjunta de todos los predictores simultáneamente.

3.2. Tipos de validación interna

Una de las principales causas de desconfianza hacía los modelos predictivos es el sobreajuste. Evaluar el rendimiento de un modelo usando los mismos datos que han sido empleados para el desarrollo de este, a menudo conduce a resultados demasiado optimistas. Con el objetivo de obtener una estimación más realista del rendimiento futuro del modelo en nuevos datos se han desarrollado diferentes métodos de validación interna. Desde la división del conjunto de datos, pasando por la validación cruzada, hasta los métodos de remuestreo bootstrapping.

A continuación se detallan los diferentes métodos de validación interna de los modelos de regresión logística y los comandos y paquetes disponibles en los programas estadísticos.

3.2.1. Validación aparente

Consiste en evaluar el rendimiento del modelo usando los mismos datos que han sido utilizados en el desarrollo del mismo, conocido como rendimiento aparente. Este método usa todos los datos disponibles para la creación y validación del modelo, pero tiene el gran problema del sobreajuste. Al utilizar los mismos datos tanto para el desarrollo del modelo como para su validación, la estimación del rendimiento del modelo será optimista. Cuando el tamaño de la muestra es enorme (>100.000 eventos) y el número de predictores no es excesivo (<100 predictores) el optimismo del modelo es insignificante. [16] Salvo en esta situación, reportar únicamente el rendimiento aparente del modelo es insuficiente, y debe ser apoyado por alguna de las técnicas de validación interna que se exponen a continuación.

Todos los programas estadísticos tienen implementados en sus paquetes básicos comandos para obtener el rendimiento aparente de los modelos, tanto en términos de discriminación como de calibración:

- **Stata**

Stata tiene implementado una serie de comandos post-estimación que permiten evaluar el rendimiento de un modelo predictivo. Estos comandos deben ser ejecutados después de que el modelo de regresión logística haya sido ajustado con los comandos *logistic* o *logit.*

El comando *lroc* permite estimar el área bajo la curva ROC y la gráfica. El test de Hosmer-Lemeshow para evaluar la calibración se puede calcular mediante el comando *estat gof*. Además de estos comandos implementados en Stata, hay otros comandos desarrollados por los usuarios que permite evaluar el rendimiento aparente del modelo. Uno de estos comandos es *Calibrationbelt*, que genera un gráfico de calibración y calcula el valor de la prueba estadística asociada para modelos con outcome binarios. Se puede usar de dos formas: Primero, identificando la variable respuesta y la probabilidad predicha. En este caso, el usuario deberá indicar si las predicciones han sido ajustadas en el conjunto de datos – validación interna – o si se trata de una evaluación en datos independientes – validación externa –. Segundo, el comando puede ser ejecutado después de ajustar un modelo de regresión logística (usando *logit* o *logistic*). En este caso, por defecto, la validación es interna (figura 4.a).

- **R**

En el software R la librería *ROCR* permite estimar el área bajo la curva ROC y obtener la gráfica. Para la calibración el paquete *ResourceSelection* tiene implementado el test de Hosmer-Lemeshow con la función *hoslem.test*. Además, existen numerosos paquetes que tienen implementadas funciones que permiten obtener los gráficos de calibración: *givitiR*, *ModelGood* o *calibrate* (figura 4.b).

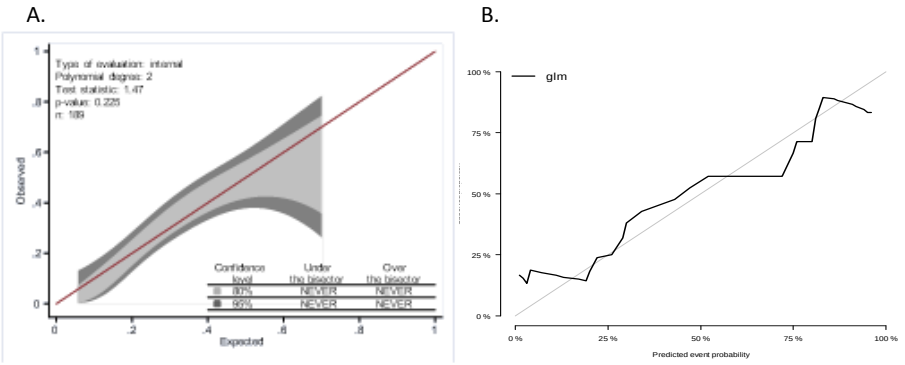

Figura 4. Gráficos de calibración de comandos creados por los usuarios para Stata y R. A). Comando calibrationbelt de Stata y B). Comando catplot2 del paquete ModelGood de R.

3.2.2. Validación por división de datos (Split-sample validation)

Es el método más simple de validación interna. Consiste en dividir de forma aleatoria la muestra original en dos submuestras, una donde se desarrolla el modelo – training – y otra donde se valida el modelo – test –. El porcentaje de la muestra original que se destina para el desarrollo del modelo y el que se aparta para la validación no está estandarizado, es habitual encontrarse divisiones 50:50 o 70:30.

Este método de validación tiene varias debilidades. Es ineficiente debido a que no utiliza todos los datos disponibles para el desarrollo del modelo, y las diferentes divisiones que se hagan de la muestra original conducirán a diferentes resultados, siendo más evidente cuanto menor es el tamaño muestral. En un reciente estudio de simulación, Ewout W. Steyerberg [22] mostró que cuando el tamaño de muestra del conjunto de validación es pequeño el resultado de la evaluación del rendimiento del modelo no es fiable. Incluso cuando el rendimiento real del modelo es 0.7, en un porcentaje significativo de muestras de validación podríamos observar un rendimiento perfecto, y poco creíble, del modelo (AUC = 1). Esta técnica, por tanto, exige que el tamaño de la muestra sea grande, pues la muestra de validación debe ser relativamente amplia. En

respuestas binarias la muestra de validación debería tener al menos 100 sujetos en la categoría menos frecuente del resultado, evento o no evento.[23] Además, los subconjuntos de datos – training y test – dado que se extraen de la misma muestra serán muy similares entre sí, por tanto, la estimación del rendimiento del modelo seguirá siendo optimista. [24] Hay varias alternativas para mejorar la división aleatoria de la muestra original evitando que los subconjuntos de desarrollo y validación del modelo sean tan similares. Cuando el tamaño de la muestra es suficientemente amplio se ha propuesto dividir la muestra por el tiempo – validación temporal – o por la localización – validación geográfica –. [25]

Cuando el tamaño de la muestra de validación no es suficiente grande esta metodología de validación es desaconsejada, debido a que la validación independiente resultará errónea y, por tanto, no debe considerarse como un paso más en la evaluación del modelo. [26]

A continuación se muestra de forma gráfica como es procedimiento de la validación mediante división de los datos.

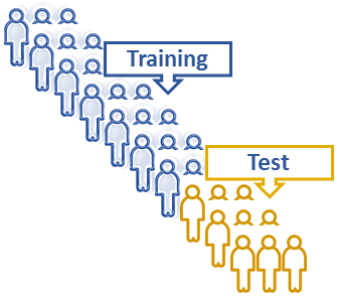

Figura 5. Split-sample (70:30) validation

Los softwares estadísticos no tienen implementada ninguna función que permita realizar este método de validación de forma automatizada. Sin embargo, se puede hacer de forma manual sin grandes conocimientos en programación. Lo más habitual es dividir el conjunto de datos en dos subconjuntos de forma aleatoria – o según el tiempo o localización en la validación temporal o geográfica –, y a través de las opciones de filtros se utiliza un subconjunto para desarrollar el modelo y el otro para validarlo.

3.2.3. Validación cruzada (Cross validation)

La validación cruzada – cross validation – es una extensión de la validación por división de datos que permite reducir el sesgo y la variabilidad en la estimación del rendimiento del modelo. [7]

La forma básica de validación cruzada consiste en dividir el conjunto de datos en dos partes del mismo tamaño. Con el método *split-sample* el modelo se desarrolla en un grupo y se valida en el otro, en este caso el modelo se desarrolla empleando todos los datos disponibles. Y posteriormente, se valida cruzando los grupos de desarrollo y validación, de modo que cada dato pueda ser validado una vez.

Por tanto, se desarrolla el modelo empleando todos los datos disponibles, posteriormente, se segmenta el conjunto de datos en dos grupos de igual tamaño. Se plantea el modelo en uno de ellos (training) y se evalúa el rendimiento en el restante (validation). A continuación, se repite el proceso intercambiando la función de los subconjuntos (figura 6). De modo que todos los datos han sido utilizados en una ocasión para la validación. El rendimiento del modelo se obtiene como el promedio del rendimiento en cada subconjunto de validación. [27]

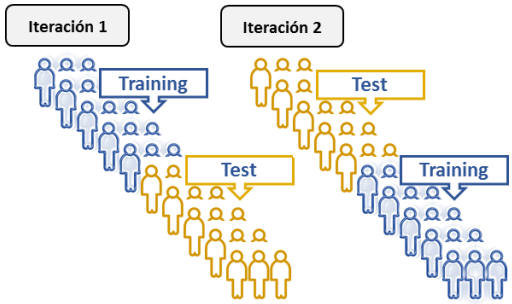

Figura 6. Cross validation

Este método de validación cruzada tiene su extensión en el *k-folds cross validation.* Consiste en dividir o segmentar el conjunto de datos en *k* grupos de igual (o similar) tamaño. Y se sigue la misma metodología que se ha explicado en el método de cross-validation básico. Se utilizan *k-1* subconjuntos para el desarrollo del modelo y el subconjunto restante se utiliza como grupo de validación. Se repite este procedimiento *k* veces, hasta que cada uno de los subconjuntos ha sido utilizado una vez como grupo de validación. Finalmente, obtenemos la medida de rendimiento como el promedio de las *k* medidas calculadas.

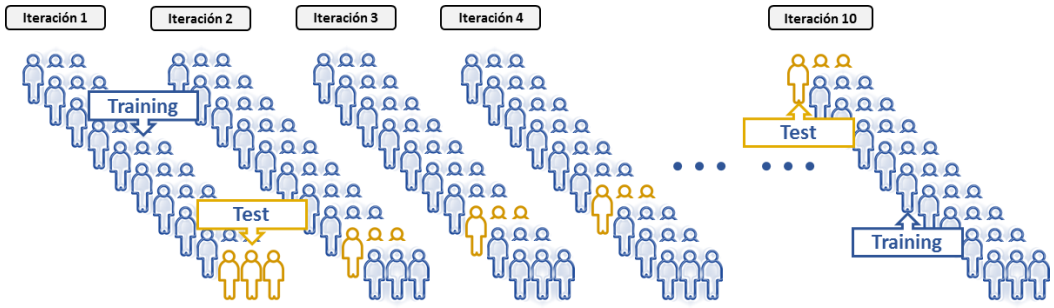

Figura 7. 10-Folds cross validation

Para obtener una estimación del rendimiento más fiable es preferible disponer de gran cantidad de estimaciones. Con el método e *k-fold cross-validation* disponemos de *k* estimaciones. Una forma comúnmente usada para incrementar el número de estimaciones es el método *Repeat k-Fold cross-validation*. Consiste en repetir en múltiples ocasiones el *k-fold cross-validation*. Los datos son reagrupados y re-estratificados en cada ronda.[27]

Los paquetes básicos de los softwares estadísticos no suelen incluir funciones para la validación cruzada pero hay varios comandos creados por los usuarios disponibles:

- **Stata**

En Stata el comando de usuario *Crossfold* realiza la validación cruzada en *k-fold* de un modelo específico. Este procedimiento divide aleatoriamente los datos en *k* particiones, luego para cada partición se ajusta el modelo especificado utilizando los otros *k-1* grupos y utiliza los parámetros resultantes para predecir la variable dependiente en el grupo no utilizado. Por defecto la métrica que utiliza para evaluar la capacidad predictiva del modelo es el error cuadrático medio (RMSE). También permite evaluar el R^2. Sin embargo, una debilidad que tiene es que no calcula el área bajo la curva ROC, estadístico más utilizado para validar los modelos.

- **R**

Calibrate emplea técnicas de validación cruzada o bootstrapping para obtener estimaciones (corregidas por el sobreajuste) de los valores predichos por el modelo versus los valores observados por grupos de riesgo. Muestra el gráfico de calibración aparente y ajustado por el optimismo.

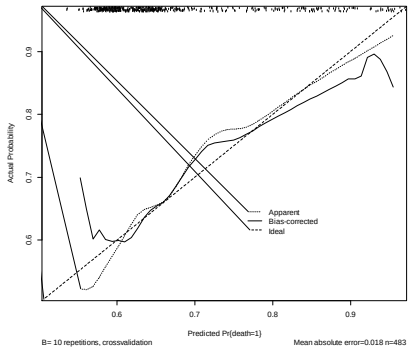

Figura 8. Gráfico de calibración usando el método de validación 10-fold cross-validation del paquete calibrate de R.

3.2.4. Validación bootstrap

Bootstrap se refiere a una técnica de simulación propuesta hace cuatro décadas por Efron y colegas.[28] Consiste en generar observaciones a partir de la distribución de la muestra original de pacientes disponible. En cada simulación se obtiene una muestra típicamente del mismo tamaño que la muestra original. La muestra simulada se genera mediante un proceso de selección aleatorio de individuos de la muestra original. Esta selección se realiza con reemplazamiento, es decir, en cada paso de la simulación cada individuo del conjunto de datos es elegible independientemente de si ha sido seleccionado o no en un paso anterior. Por tanto, en cada muestra bootstrap algunos de los individuos pueden estar representados varias veces y otros pueden no aparecer ninguna. Podría darse el hipotético caso que una muestra bootstrap estuviera compuesta por una población representada por un mismo individuo *n* veces, aunque claro está que la probabilidad es muy próxima a cero.

Esta muestra aleatoria con reemplazamiento es repetida cientos o miles de veces para generar nuevas muestras simuladas y asegurar estadísticos precisos.

Uno de los principales uso de las técnicas bootstrap consiste en la validación interna de modelos de regresión. [29] La ventaja de la validación bootstrap es que para la construcción del modelo emplea todos los datos disponibles, lo que permitirá obtener ecuaciones de regresión más robustas. Esto es todavía más importante cuando tenemos tamaños de muestra pequeños.

Como técnica de validación interna, la ventaja del bootstrapping, respecto a la validación cruzada, es que los efectos de las estrategias de selección de los predictores en la construcción del modelo, y por tanto el grado de sobreajuste y optimismo del

modelo, pueden cuantificarse repitiendo el proceso de selección del predictor en cada muestra bootstrap. Además, el bootstrapping nos proporciona una estimación del llamado factor de ajuste o corrección (shrinkage factor), por el cual el modelo (los coeficientes de regresión) y sus medidas de rendimiento pueden ser contraídos y de ese modo ajustar por sobreajuste.

El sobreajuste, el optimismo y la mala calibración también pueden abordarse y justificarse mediante la aplicación de procedimientos de reducción o penalización (shrinkage) [21][30][31]

El optimismo es un problema ampliamente conocido de los modelos predictivos. El rendimiento del modelo en nuevos pacientes a menudo es peor que el rendimiento estimado en el conjunto de datos en el que se desarrolla el modelo, el ya mencionado "rendimiento aparente".

Por último, las técnicas bootstrap permiten evaluar la fiabilidad de un factor de riesgo. Frecuentemente en cada análisis de regresión realizado en las diferentes muestras bootstrap simuladas obtendremos modelos diferentes que incluirán predictores distintos. Habrá predictores que no serán incluidos nunca en el modelo y otros que aparezcan en todas las ocasiones. Usualmente predictores que resultan significativos en más del 50% de las muestras bootstrap pueden ser considerados predictores fiables y deberían ser incluidos en el modelo de regresión final.[29]

Las técnicas de validación bootstrap incluyen los siguientes pasos:[7]

1. Desarrollar el modelo predictivo usando todos los datos disponibles y calcular el rendimiento aparente.
2. Generar *n* muestras bootstrap con reemplazamiento desde la muestra original y del mismo tamaño que esta.
3. Desarrollar un modelo predictivo en la primera muestra bootstrap aplicando las mismas estrategias empleadas en el desarrollo del modelo original.
4. Determinar el rendimiento aparente del modelo en la muestra bootstrap, lo que se conoce con el nombre de rendimiento bootstrap.
5. Determinar el rendimiento del modelo bootstrap en la muestra original, llamado test de rendimiento. Es decir, aplicar el modelo que se ha obtenido en la muestra bootstrap en los datos originales.

6. Calcular el optimismo como la diferencia entre el rendimiento bootstrap (paso 4) y el test de rendimiento (paso 5).
7. Repetir los pasos 4, 5 y 6 en cada una de las *n* muestras bootstrap, al menos 100 muestras.
8. Obtener la estimación del rendimiento ajustada por el optimismo restando al rendimiento aparente (calculado en el paso 1) la media del optimismo obtenido en las *n* muestras bootstrap.

Es extremadamente importante que todos los aspectos del ajuste del modelo se incorporen exactamente igual en cada muestra aleatoria o bootstrap (paso 3), incluyendo las estrategias de selección de variables, las transformaciones y las interacciones. Omitir estos pasos es común pero puede dirigir a evaluaciones sesgadas del ajuste. Reajustar usando los mismos predictores en cada muestra Bootstrap, salvo que el modelo se construyera usando todos los predictores, no es un método válido.

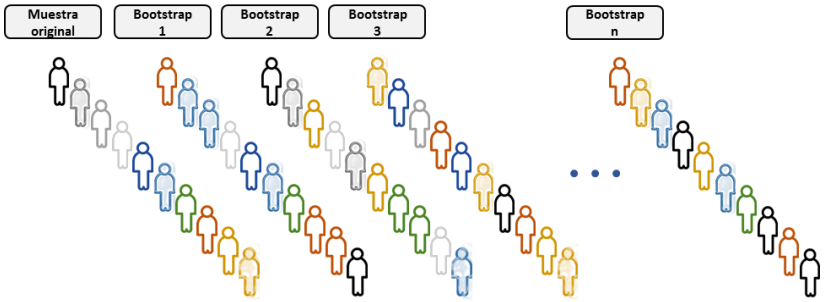

Figura 9. Bootstrap validation

Los software estadístico tiene algunos comandos que permiten realizar la validación interna del modelo con técnicas bootstrap, principalmente desarrollados en R.

- **Stata**

swboot utiliza muestras bootstrap de tamaño N (basadas en el número de observaciones sin valores faltantes) para validar la elección de variables en procedimientos por pasos para la regresión lineal o logística; las variables seleccionadas se muestran para cada muestra extraída; un resumen al final cuenta el número total de veces que se selecciona cada variable. Es posible realizar la selección de variables hacia atrás “backward” y hacía adelante "forward".

bootstrap permite ejecutar el comando de Stata deseado, por ejemplo *logistic*, utilizando las observaciones remuestreadas con reemplazamiento.

- **R**

La función *validate* es, sin duda la función más similar al comando desarrollado en este trabajo para el software Stata. Permite calcular los estadísticos de rendimiento del modelo mediante técnicas de remuestreo bootstrapping de un modelo de regresión con o sin estrategías de selección de variables hacia atrás – backward –. Proporciona el estadísdito Dxy de Somers de correlación de rangos, el intercepto y la pendiente de una ecuación de regresión logística, la diferencia máxima absoluta entre las probabilidades pronosticadas y observadas, además de otros índices de discriminación del modelo. La pendiente corregida se puede considerar como el factor de contracción – shrinkage factor – que tiene en cuenta el sobreajuste. Una de las principales debilidades de esta función es que no reporta el área bajo la curva ROC, aunque se puede obtener a partir del estadístico Dxy de Somers.

4. Resultados

4.1. Comando validation

Se ha desarrollado una macro o comando en el software Stata que nos permite evaluar el rendimiento del modelo empleando técnicas de validación Bootstrap. A pesar de que existen algunos comandos de usuario, tanto en Stata como, sobre todo, en R que permiten validar los modelos con estas técnicas en el software Stata no hay ninguno que sea realice una validación completa.

Con el comando *validation* se pretende que el investigador pueda realizar la validación interna del modelo predictivo de forma sencilla. Para evaluar la capacidad discriminante del modelo se reporta el área bajo la curva ROC, que ya hemos mencionado que es el estadístico más utilizado entre la comunidad científica. La calibración del modelo se evalúa mediante el test de Hosmer-Lemeshow Goodness of Fit y la pendiente de calibración. Gracias al empleo de técnicas de remuestreo bootstrapping se puede estimar el grado de optimismo del rendimiento. Para evaluar la robustez o fiabilidad de los predictores incluidos en el modelo se reporta la frecuencia de cada uno de ellos en el modelo final. Finalmente, y para añadir más valor al comando desarrollado, se incluye la opción de poder generar los gráficos más habituales en la validación de modelos.

El comando *validation* es un post-comando que debe utilizarse después de ejecutar los comandos *logit* o *logistic*. En caso de no haber ajustado el modelo logístico

previamente, en la pantalla de resultados se imprimirá un mensaje de error indicando que el modelo debe ser ajustado antes. Por tanto, primero se debe ajustar el modelo como sigue:

logistic *depvar indepvars*

Box 1. Estructura del modelo de regresión logística

Una vez hemos ajustado nuestro modelo final empleado todos los datos disponibles podemos ejecutar el comando *validation*. La estructura sintáctica es la siguiente:

validation *[varlist] [if] [in], reps(integer) [dummy1(string)] [dummy2(string)] [pr(real) pe(real)] [seed(integer)] [graph] [group(integer)]*

Box 2. Estructura sintáctica del comando validation

- ***varlist*** – La lista de variables que se han evaluado pero no se han incluido en el modelo final ajustado. Es decir, las variables que se incluían en el modelo máximo y que han sido excluidas del modelo siguiendo las estrategias de desarrollo del modelo y selección de variables establecidas. Todas las sentencias que aparecen entre corchetes son opcionales. Si no se incluye ninguna variable en *varlist* el modelo ajustado en cada muestra bootstrap siempre incluiría los mismo predictores (los mismos que se han ajustado en el modelo logístico ejecutado en el paso anterior). Esta circunstancia sólo será válida si el modelo ajustado incluye todos los predictores que se han evaluado.
- ***if*** e ***in*** – Las sentencias *if* e *in* son opciones de selección de los datos sobre los que el comando trabajará. Son habituales en la mayoría de comandos implementados en Stata.
- ***reps(#)*** – Es la única opción obligatoria del comando. Indica el número de muestras bootstrap obtenidas desde la muestra original. Debe ser un número entero entre el 1 y el 300.
- ***dummy1(#)*** y ***dummy2(#)*** – Variable indicadora o dummy. Si hay variables dummy que evaluar deben ser incluidas en esta opción. Debido a que el comando *validation* no admite variables factor las variables dummies deben haberse generado previamente. El comando permite incluir un máximo de dos variables dummy.

- ***pr(#)*** – Especifica el nivel de significación para eliminar una variable del modelo. Variables con p-valor superior a *pr()* son eliminados del modelo. Por defecto el valor de significación es 0.1.
- ***pe(#)*** – Especifica el nivel de significación para añadir una variable al modelo. Variables con p-valor inferior a *pe()* son incluidos en el modelo. Por defecto el valor de significación es 0.001.
- ***seed(#)*** – Especifica la semilla de aleatorización empleada en el proceso de muestreo bootstrap. Nos permite reproducir exactamente los resultados.
- ***graph*** – Genera los gráficos de calibración y discriminación del modelo.
- ***group(#)*** – Indica el número de grupos que se emplearán para generar los gráficos de calibración y el cálculo del valor del test de Hosmer-Lemeshow. Por defecto el comando utiliza deciles de riesgo, es decir, 10 grupos.

4.1.1. ¿Cómo utilizar el comando *validation*?

El comando *validation* se utiliza después de ajustar un modelo de regresión logística. A partir del comando previo – *logit* o *logistic* – el comando recupera la variable dependiente binaria – outcome – y el resto de variables predictoras que han sido incluidas en el modelo final ajustado.

En la lista de variables que se debe incluir en el comando *validation* sólo se deben indicar aquellos potenciales predictores que han sido evaluados pero que en el proceso de ajuste del modelo han sido excluidos del modelo final. Si alguno de estos predictores es una variable dummy no se incluirá en la lista de variables sino en las opciones *dummy1()* o *dummy2()* del comando, de modo que puedan ser evaluadas globalmente. Si el modelo final ajustado incluye una variable dummy estas deben indicarse entre paréntesis.

Una vez indicadas las variables predictoras el siguiente paso es especificar el número de remuestreos que se van a utilizar en el proceso de validación. Esta es la única opción obligatoria *reps(#).* El número máximo de remuestreos que se puede realizar está limitado por el tamaño máximo de matriz que la versión de Stata permite. La versión Intercooled el máximo permitido es 800.

Además de las variables y el número de remuestreos es importante fijar la estrategia de selección de variables. Esta debe ser exactamente igual a la empleada en el ajuste de modelo en la muestra original. Para ello utilizamos las opciones *pr(#)* y *pe(#).* Si

ejecutamos el comando sin especificar estas opciones el programa tomará por defecto los valores de significación de 0.1 para la exclusión de variables y el valor 0.001 para la inclusión. Por definición el valor de inclusión debe ser menor que el valor de exclusión. Si queremos aplicar una estrategia de selección de variables hacia atrás – backward – con un nivel de significación del 5% en las opciones del comando indicaremos *pr(0.05)* y el comando *pe(#)* no será necesario. La estrategia de selección hacía adelante – forward – está siendo todavía depurada.

Si se quiere obtener unos resultados reproducibles se debe especificar la semilla de aleatorización *seed(#)*. De no ser así, dado que el comando utiliza sentencias de números aleatorios, cada vez que se ejecute el comando obtendremos resultados distintos. Por último, aparece la opción *graph()* permite generar gráficos de calibración y discriminación. La opción *group(#)* trabaja en los gráficosde calibración de calibración y en el cálculo del estadístico de Hosmer-Lemeshow. Por defecto, el comando trabaja para 10 grupos de riesgo.

4.1.2. ¿Qué hace el comando *validation*?

Como se ha comentado con anterioridad el comando validation debe ejecutarse después de haber ajustado un modelo de regresión logística con los comandos *logit* o *logistic*.

Al ejecutar *validation* la primera función del comando es guardar el modelo ejecutado en el paso inmediatamente anterior. Se almacena, por tanto, la variable dependiente, de carácter binario, y las variables predictoras incluidas en el modelo ajustado final. Se estima el rendimiento aparente del modelo. La discriminación se evalúa mediante el estadístico C o área bajo la curva ROC y la calibración mediante la prueba de Hosmer-Lemeshow. También se estima el factor de contracción heurístico propuesto por van Houwelingen y Le Cessie.

Una vez obtenido el rendimiento aparente del modelo se da el paso a las técnicas bootstrap. A partir de la muestra original se genera una muestra bootstrap con reemplazamiento del mismo tamaño que la original. Se ajusta un modelo máximo que incluye las variables predictoras incluidas en el modelo final junto con aquellas que se habían evaluado inicialmente pero fueron eliminadas en la estrategia de selección de variables, es decir, las variables incluidas en *varlist* y/o en las opciones *dummy1* y *dummy2* si han sido especificadas. A partir del modelo máximo, y siguiendo las mismas estrategias de selección de variables que se han utilizado en el ajuste del modelo

original, ajustamos el modelo final en esta muestra bootstrap. Para establecer las estrategias de selección de variables se deben especificar las opciones *pr()* y *pe()*.

Alcanzado el modelo final se obtiene el rendimiento aparente del modelo en la muestra bootstrap. Al igual que para el rendimiento aparente del modelo original, la discriminación se evalúa mediante el área bajo la curva ROC y la calibración por medio de la prueba de Hosmer-Lemeshow. Esto se conoce como el rendimiento bootstrap.

A continuación, se recupera la muestra original y, a partir del modelo ajustado en la muestra bootstrap, se calculan las probabilidades predichas para cada individuo. Ahora, en la muestra original, se evalúa la discriminación nuevamente mediante el área bajo la curva ROC y la calibración, en esta ocasión a través de la pendiente de calibración, ejecutando un modelo de regresión logística que incluya como única variable predictora las predicciones generadas por modelo ajustado en la muestra bootstrap. De este modo se podrá estimar el shrinkage factor. Esta evaluación se conoce como test de rendimiento. Finalmente, se calcula el optimismo del modelo como la diferencia entre el rendimiento bootstrap y el test de rendimiento.

Se vuelve a generar una nueva muestra bootstrap y se repite el proceso el número de veces indicado en la opción *reps(#)*.

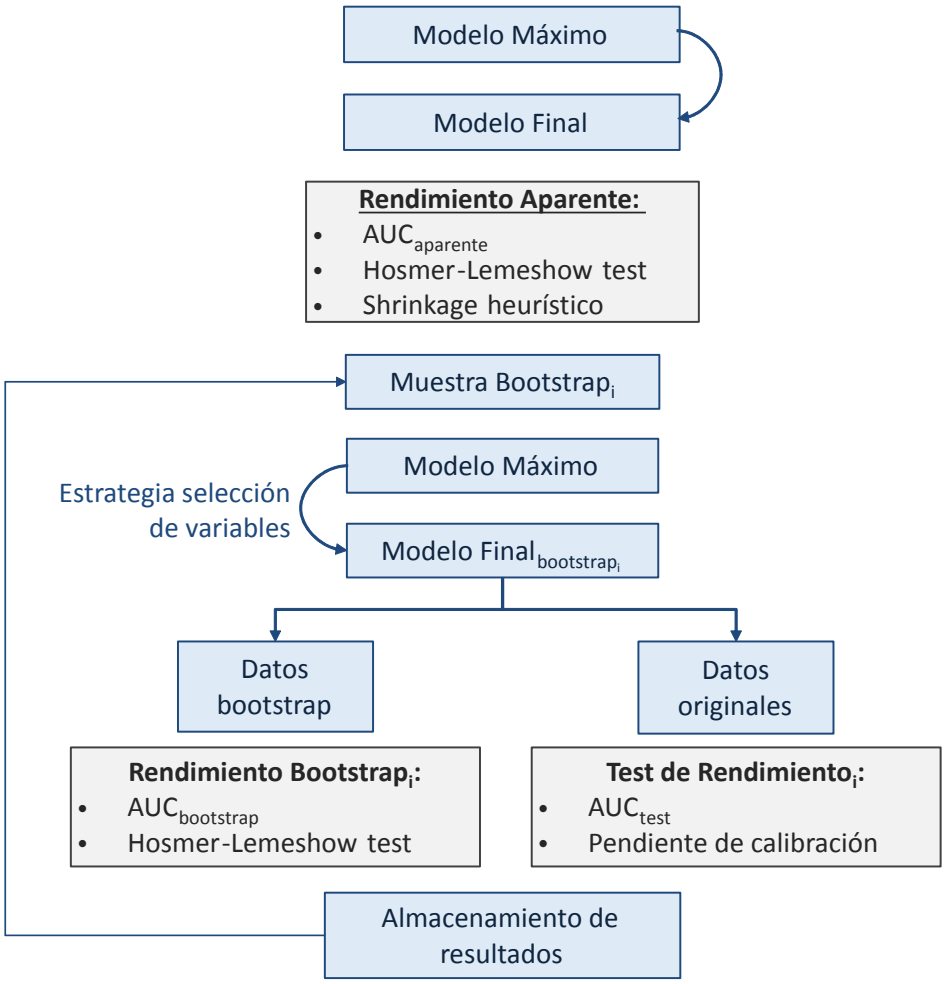

Figura 10. Algoritmo del comando validation

4.1.3. Resultados del comando *validation*

La primera sección de la salida del comando validation reporta los modelos de regresión logística. El modelo máximo que incluye todas las variables predictoras que han sido evaluadas y el modelo final que incluye los predictores que han cumplido los criterios especificados en la selección de variables.

El formato de salida de los modelos es el mismo que emplean el comandos de modelos de regresión logística implementado en stata (comando *logistic*). Los modelos aparecen etiquetados como *Maximum model* y *The final model* respectivamente. En la cabecera del modelo aparecen los estadísticos globales, y para cada variable incluida en el modelo se reporta el odds ratio, error estándar, el estadístico z y su p-valor asociado y el intervalo de confianza al 95% para odds ratio.

La segunda sección de los resultados muestra los primeros resultados de la validez del modelo, el rendimiento aparente del modelo final. Tanto en términos de capacidad

discriminatoria, mediante el área bajo la curva ROC y su intervalo de confianza al 95%, como en términos de calibración o capacidad predictiva, mediante el test de Hosmer-Lemeshow (se reporta tanto el estadístico como el p-valor asociado). Finalmente se muestra un estadístico que cuantifica el sobreajuste del modelo, el shrinkage heurístico.

La última sección de la salida está dedicada a los resultados de la validación interna empleando las técnicas bootstrap. En la cabecera de la sección se indica el número de remuestreos realizados. Se muestra la estimación del optimismo y el área bajo la curva ROC ajustado por este con su intervalo de confianza al 95%. Seguidamente, se reporta el shrinkage factor y el número de veces que el test de Hosmer-Lemeshow ha resultado estadísticamente significativo (p-valor < 0.05) en las muestras bootstrap, indicativo de que el modelo no estaría bien calibrado. Dado el creciente interés en el uso de técnicas bootstrap para el desarrollo de los modelos predictivos, como complemento de los resultados se muestra el número de veces que cada predictor ha sido incluido en el modelo final. Es habitual considerar que un predictor es consistente o fiable cuando se incluyó en más de la mitad de las muestras bootstrap.

La salida del comando se completa con la posibilidad de obtener los gráficos de discriminación y calibración del modelo (opción *graph*). El comando reporta un gráfico conjunto que incluye la curva ROC y un histograma de las probabilidades predichas por el modelo en función de la variable dependiente (discriminación) y un gráfico de calibración que cruza las probabilidades predichas por el modelo versus probabilidades observadas para los distintos grupos de riesgo (calibración). La opción *group(#)* permite definir el número de grupos de riesgo que se quiere realizar.

Resultados comando validation

- Rendimiento aparente del modelo final
  - $AUC_{aparente}$
  - Prueba de Hosmer-Lemeshow
  - Shrinkage heurístico
- Validación bootstrap
  - Optimismo

$$\sum_1^n \frac{AUC_{Bootstrap} - AUC_{Test}}{n}$$

  - $AUC_{ajustado}$

$$AUC_{aparente} - Optimismo$$

  - Shrinkage factor

$$\sum_1^n \frac{Pendiente}{n}$$

Siendo n el número de remuestreos realizados

  - Número y porcentaje de test Hosmer-Lemeshow estadísticamente significativos (p-valor < 0,05)
  - Frecuencia de cada predictor en el modelo final
- Gráficos
  - Curva ROC
  - Histograma de las probabilidades predichas
  - Gráfico de calibración

Box 3. Resultados que se muestran en la salida del comando validation

4.2. Aplicación del comando validation en una serie de casos

Un conjunto de datos utilizado en la documentación de Stata fue seleccionado para demostrar cómo aplicar el comando validation.

El conjunto de datos lbw.dta es una base que contiene datos de 189 mujeres con información de una serie de potenciales predictores de bajo peso en los recién nacidos. Para aplicar el comando validation a un ejemplo práctico se pretende establecer un modelo predictivo de bajo peso al nacer, definido como el peso del bebé inferior a 2.500 gramos.

La variable de interés – variable dependiente – que se desea evaluar es **low**. Variable binaria que toma valores:

- 0. - Peso del bebé ≥ 2.500 gramos.
- 1. - Peso del bebé < 2.500 gramos.

Las características de las madres que se han recogido y que son analizadas como potenciales predictores de bajo peso al nacer son las siguientes:

- **age.** Edad de la madre.
- **lwt.** Peso de la madre en el último periodo menstrual.
- **race.** Raza de la madre. Toma los valores:
  - 0. - Raza blanca.
  - 1. - Raza negra.
  - 2. - Otras razas.
- **smoke.** Fumadora durante el embarazo.
- **ptl.** Historia de partos prematuros. (número de partos)
- **ht.** Historía de hipertensión.
- **ui.** Presencia de irritabilidad uterina.
- **ftv.** Número de visitas al médico durante el primer trimestre del embarazo.

```
Contains data                                  Hosmer & Lemeshow data
  obs:           189                           15 Jan 2016 05:01
 vars:            11
 size:         3,402
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
id              int     %8.0g                 identification code
low             byte    %8.0g                 birthweight<2500g
age             byte    %8.0g                 age of mother
lwt             int     %8.0g                 weight at last menstrual period
race            byte    %8.0g      race       race
smoke           byte    %9.0g      smoke      smoked during pregnancy
ptl             byte    %8.0g                 premature labor history (count)
ht              byte    %8.0g                 has history of hypertension
ui              byte    %8.0g                 presence, uterine irritability
ftv             byte    %8.0g                 number of visits to physician during 1st trimester
bwt             int     %8.0g                 birthweight (grams)
-------------------------------------------------------------------------------
Sorted by:
```

Hay 59 (31.2%) bebés con bajo peso al nacer.

Con el objetivo de evaluar la aplicación del comando validation se ha agrupado la variable número de visitas al médico durante el primer trimestre en tres categorías: ninguna visita, una visita y dos o más visitas.

Se ha ajustado un modelo predictivo de regresión logística que incluía todos los potenciales predictores recogidos en la base de datos – Modelo máximo –. Se ha empleado una estrategia de selección de variables hacia atrás – backward – hasta

alcanzar el modelo final. Las variables con p-valor superior al nivel de significación 0.05 en el análisis multivariante son excluidas del modelo predictivo final.

Dado que el comando validation no admite variables factor, se ha creado de forma manual las variables dummy (raza y número de visitas al médico).

```
logistic low age lwt (race2 race3) smoke ptl ht ui (ftv2 ftv3)
```

Box 4. Sintaxis modelo máximo en Stata

```
Logistic regression                             Number of obs     =        189
                                                LR chi2(10)       =      34.05
                                                Prob > chi2       =     0.0002
Log likelihood = -100.31322                     Pseudo R2         =     0.1451
```

low	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
age	.9733783	.0364162	-0.72	0.471	.9045579	1.047435
lwt	.9844039	.0068528	-2.26	0.024	.971064	.9979272
race2	3.502573	1.854044	2.37	0.018	1.241122	9.884624
race3	2.273543	1.025358	1.82	0.069	.9393235	5.502896
smoke	2.37565	.9845148	2.09	0.037	1.054456	5.352253
ptl	1.830687	.6543382	1.69	0.091	.9085969	3.688562
ht	6.643481	4.689448	2.68	0.007	1.665544	26.49935
ui	2.080752	.9580388	1.59	0.112	.8439221	5.130247
ftv2	.7416319	.3440397	-0.64	0.519	.2987582	1.841013
ftv3	1.195773	.5390299	0.40	0.692	.4942472	2.89303
_cons	1.800509	2.184937	0.48	0.628	.1668988	19.42395

Note: **_cons** estimates baseline odds.

Siguiendo la estrategia de selección de variables se han excluido del modelo las siguientes variables:

- Paso 1. - ftv (p-valor = 0.667). Al tratarse de una variable dummy se debe evaluar las dos categorías globalmente. Stata permite realizar este test mediante el comando *testparm*.
- Paso 2. - age (p-valor = 0.457).
- Paso 3. - ptl (p-valor = 0.140).

El resto de variables se mantienen significativas (p-valor < 0.05) y, por tanto, son incluidas en el modelo predictivo final de bajo peso al nacer. El modelo incluye el peso de la madre en el último periodo menstrual, la raza, el estatus de fumadora durante el embarazo, la historia de hipertensión y la presencia de irritación uterina.

```
logistic low lwt (race2 race3) smoke ht ui
```

Box 5. Sintaxis modelo final en Stata

```
Logistic regression                               Number of obs     =        189
                                                  LR chi2(6)        =      30.43
                                                  Prob > chi2       =     0.0000
Log likelihood = -102.11978                       Pseudo R2         =     0.1297
```

low	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
lwt	.9834361	.0066887	-2.46	0.014	.9704134	.9966336
race2	3.758631	1.959795	2.54	0.011	1.352705	10.44375
race3	2.526023	1.087054	2.15	0.031	1.08675	5.871446
smoke	2.817403	1.105908	2.64	0.008	1.305356	6.080917
ht	6.490237	4.483259	2.71	0.007	1.676009	25.13302
ui	2.471801	1.106213	2.02	0.043	1.028189	5.942297
_cons	1.054066	.9884219	0.06	0.955	.1677556	6.623063

Note: **_cons** estimates baseline odds.

Una vez ajustado el modelo final se puede ejecutar el comando *validation*.

Cuando se ejecuta el comando *logistic* con el modelo final, es importante que la variable dummy incluida en el modelo (raza) aparezca entre paréntesis. De no ser así el comando *validation* las considerará como variables independientes y, por consiguiente, sus categorías no serán evaluadas globalmente.

Se ha llevado a cabo la validación bootstrap siguiendo la siguiente sintaxis:

```
validation ptl age, dummy1(ftv2 ftv3) reps(100) seed(159) pr(0.05) graph group(5)
```

Box 6. Sintaxis comando validation en Stata

Los candidatos predictores que en el proceso de selección de variables han sido excluidos del modelo son indicados en *varlist*. Como *ftv* se agrupó en tres categorías se debe especificar como variable dummy [*dummy1(ftv2 ftv3)*]. El comando realiza 100 muestras bootstrap [*reps(100)*] y se ha fijado una semilla de aleatorización para poder reproducir los análisis [*seed(159)*]. Dado que la estrategia de selección de variables consistía en ir eliminando los potenciales predictores con p-valor mayor a 0.05, se especifica en la opción *pr(0.05)*. Finalmente para solicitar los gráficos se incluye la opción *graph*. La opción *group(5)* se especifica para establecer cinco grupos de riesgo para el gráfico y test de calibración.

La salida de resultados del comando es la siguiente:

1. Modelo máximo. Incluye las variables ajustadas en el modelo final (lwt, smoke, ht, ui y race) más las variables que fueron excluidas en el proceso de ajuste (ptl, age y ftv).

Model **maximum** (*The maximum model*)

Logistic regression — Number of obs = 189; LR chi2(10) = 34.05; Prob > chi2 = 0.0002; Log likelihood = -100.31322; Pseudo R2 = 0.1451

low	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
lwt	.9844039	.0068528	-2.26	0.024	.971064	.9979272
smoke	2.37565	.9845148	2.09	0.037	1.054456	5.352253
ht	6.643481	4.689448	2.68	0.007	1.665544	26.49935
ui	2.080752	.9580388	1.59	0.112	.8439221	5.130247
ptl	1.830687	.6543382	1.69	0.091	.9085969	3.688562
age	.9733783	.0364162	-0.72	0.471	.9045579	1.047435
race2	3.502573	1.854044	2.37	0.018	1.241122	9.884624
race3	2.273543	1.025358	1.82	0.069	.9393235	5.502896
ftv2	.7416319	.3440397	-0.64	0.519	.2987582	1.841013
ftv3	1.195773	.5390299	0.40	0.692	.4942472	2.89303
_cons	1.800509	2.184937	0.48	0.628	.1668988	19.42395

Note: **_cons** estimates baseline odds.

2. Modelo final. Incluye las cinco variables (race es una variable dummy) especificadas en el modelo final.

Model **final** (*The final model*)

Logistic regression — Number of obs = 189; LR chi2(6) = 30.43; Prob > chi2 = 0.0000; Log likelihood = -102.11978; Pseudo R2 = 0.1297

low	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
lwt	.9834361	.0066887	-2.46	0.014	.9704134	.9966336
race2	3.758631	1.959795	2.54	0.011	1.352705	10.44375
race3	2.526023	1.087054	2.15	0.031	1.08675	5.871446
smoke	2.817403	1.105908	2.64	0.008	1.305356	6.080917
ht	6.490237	4.483259	2.71	0.007	1.676009	25.13302
ui	2.471801	1.106213	2.02	0.043	1.028189	5.942297
_cons	1.054066	.9884219	0.06	0.955	.1677556	6.623063

Note: **_cons** estimates baseline odds.

3. Rendimiento aparente del modelo final.

```
Apparent performance
------------------------------------------------------------------------------
                  ROC area =    0.735   95%CI (0.660-0.810)
Hosmer-Lemeshow chi2(3) =    1.997   Prob > chi2 = 0.573
    Heuristic shrinkage =    0.706
------------------------------------------------------------------------------
```

El aspecto discriminante del modelo es evaluado mediante el área bajo la curva ROC. La capacidad discriminante del modelo es moderadamente buena AUC = 0.735 (0.66; 0.81).

El apartado de la calibración es evaluado mediante el test de Hosmer-Lemeshow. Tiene tres grados de libertad porque se ha especificado que se hicieran cinco grupos de riesgo. Un p-valor 0.573 > 0.05 indica que no existen diferencias estadísticamente significativas entre el número de eventos esperados según el modelo para cada grupo de riesgo respecto a los observados.

Por último, aparece el shrinkage heurístico que evalúa el grado de sobreajuste. En este caso un valor de 0.706, relativamente lejos de la unidad, indica que el modelo está sobreajustado. En efecto, como se ha comentado en este trabajo, una recomendación bastante usada en el desarrollo de modelos predictivos es utilizar un ratio de 1/10 para decidir el número de variables a estudiar respecto al número de eventos (o no eventos). En el ejemplo aplicado, hay 59 bebés de bajo peso, por lo que siguiendo esta metodología no deberían ser analizados más de 6 candidatos predictores. Sin embargo el modelo máximo incluye hasta 10.

4. Validación bootstrap del modelo final.

```
Bootstrap                       Number of replications: 100
------------------------------------------------------------------------------
                 Optimism =    0.060
    ROC area (adjusted) =    0.675   95%CI (0.600-0.751)
    Bootstrap shrinkage =    0.666
HL test significatives:
                  Number =    18
              Proportion =    18.0%   95%CI (11.0%-26.9%)
------------------------------------------------------------------------------
```

```
Number of replications: 100
Summary: (Number of times each variable is selected)
--------------------------------------------------------------------------------
    lwt:                    68
    smoke:                  60
    ht:                     73
    ui:                     47
    ptl:                    44
    age:                    10
    race2:                  73
    race3:                  73
    ftv2:                   11
    ftv3:                   11
--------------------------------------------------------------------------------
```

En la cabecera se muestra el número de repeticiones (muestras bootstrap) llevadas a cabo, en este caso son 100. A continuación, se muestra el optimismo del modelo (optimismo = 0.06) y el área bajo la curva ROC ajustado por el optimismo. El bootstrap shrinkage es la pendiente de calibración, un valor de 0.666 indica que el modelo está sobreajustado. Un modelo bien ajustado debería tener una pendiente de calibración próxima a la unidad. Seguidamente se muestra el número de test de Hosmer-Lemeshow que han resultado estadísticamente significativos (síntoma de una mala calibración). En el ejemplo de aplicación en 18 de las 100 muestras bootstrap en las que se ajustó el modelo han resultado estadísticamente significativas. Es decir, el grado de acuerdo entre las predicciones realizadas por el modelo y lo observado era malo.

Para finalizar el apartado de resultados de la metodología bootstrap se indica la frecuencia de aparición en el modelo final de cada uno de los potenciales predictores analizados. Los predictores que en más ocasiones han sido incluidos en el modelo final son la historia de hipertensión (ht: 73 veces), la raza de la madre (race: 73), el peso de la madre en el último periodo menstrual (lwt: 68) y el status de fumadora (smoke: 60). Podemos comprobar que tanto race (73 veces) como ftv (11 veces) siempre que han sido incluidas en el modelo han sido incluidas globalmente, no puede aparecer una categoría y otra no. De las cinco variables incluidas en nuestro modelo predictivo final, a excepción de la presencia de irritabilidad uterina (ui: 47 veces) que en el modelo original tenía el p-valor más próximo a nivel de significación exigido, el resto han sido incluidas en el ajuste final en más del 60% de los casos. Esto indica que son predictores confiables de bajo peso al nacer.

5. Gráficos.

Para completar la salida de resultados del comando *validation* se incluye la parte gráfica. En el caso de no haber especificado la opción *graph* esta parte no se ejecuta.

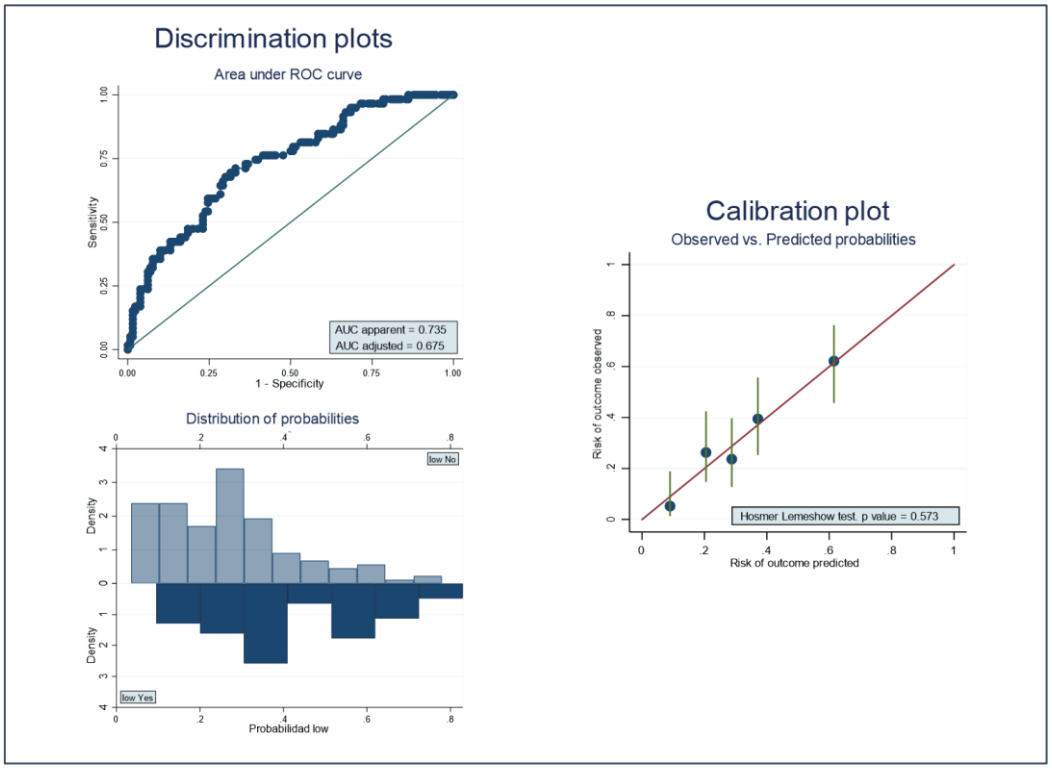

Figura 11. Gráficos de discriminación y calibración reportados por el comnado validation.

Se trata de un gráfico obtenido de la combinación de varios. En la parte izquierda se muestran los gráficos de discriminación. En la parte superior, la curva ROC, donde se especifica el rendimiento aparente ($AUC_{apparent}$ = 0.735) y el rendimiento ajustado por el optimismo ($AUC_{adjusted}$ = 0.675). En la parte inferior se muestran las distribuciones de las probabilidades de tener un bebé con bajo peso en función del evento. Hacía arriba las probabilidades de las madres que no tuvieron un bebé de bajo peso y hacía abajo las probabilidades de las madres que sí tuvieron un bebé de bajo peso. Se puede apreciar que las madres con bebés de bajo peso tienen las probabilidades más concentradas en los valores bajos. Si la capacidad discriminante del modelo fuera mejor estos gráficos serían más claros visualmente.

En la parte derecha del gráfico se muestra el gráfico de calibración. Se crean grupos de riesgo (en el ejemplo cinco) con las probabilidades predichas por el modelo original. Y se compara el número de eventos esperados en cada grupo con el número de evento observados. Si la calibración fuera perfecta los puntos caerían en la recta de pendiente 1.

5. Fortalezas y debilidades

5.1. Fortalezas

El principal fortaleza del comando *validation* es que permite realizar la validación interna de modelos predictivos mediante el método más robusto y recomendado, el bootstrap.

Su sencillez a la hora de utilizarlo también puede resultar muy atractiva al usuario. Pues no necesita de unos conocimientos avanzados en programación, ni tan siquiera en el software estadístico Stata. Tan sólo hace falta completar una serie de opciones básicas para obtener unos resultados de validación a través de técnicas contrastadas.

Otra fortaleza del comando es lo completo de los resultados. A diferencia de los comandos ya existentes (los más completos en el software R), el comando *validation* combina los resultados numéricos con los resultados gráficos. Hasta ahora la única forma de obtener ambos era empleando diferentes comandos.

5.2. Debilidades

Entre las debilidades del comando destaca que el número de variables dummy incluidas en el modelo está restringido. El comando *validation* trabaja con un máximo de tres variables dummy. Una de ellas puede ser incluida en el modelo final ajustado entre paréntesis (como en el ejemplo *race2 race*3), y fuera del modelo final se pueden incluir otras dos variables dummy en las opciones del comando, *dummy1(#)* y *dummy2(#)*.

Otra debilidad del comando es la restricción en el número de repeticiones que se pueden realizar, El número máximo es 800 repeticiones debido a que el comando trabaja con matrices. Por defecto, el tamaño máximo permitido en la versión Stata 15/IC es 400, pero se puede ampliar hasta 800 con el comando *set matsize*.

Otra debilidad importante del comando *validation* es el tiempo de ejecución. Mientras que los comando como *calibrate* de R son muy eficientes y obtenemos los resultados en apenas segundos (dependerá del número de observaciones y variables que estemos utilizando), el comando *validation* necesita más tiempo para reportar los resultados.

Podría considerarse otra debilidad, que hay comandos que reportan más estadísticos o índices. El objetivo para el desarrollo del comando fue con la idea de sencillez y atractivo. Por lo que se ha considerado que incluir más estadísticos o índices en la salida

de resultados puede resultar contraproducente. Por ello, el comando se ha centrado en validar los aspectos importantes discriminación y calibración de forma clara y concisa.

6. Conclusiones

Una de las más importantes deficiencias de los modelos predictivos que se desarrollan en el ámbito biomédico es la carencia de validación. Este comando, validation, permitirá a los usuarios de Stata realizar la validación interna de los modelos predictivos de regresión logística, empleando técnicas de remuestreo bootstrap. Reportando resultados numéricos y gráficos del rendimiento del modelo tanto en términos de discriminación como de calibración.

7. Futuro

Afortunadamente, y dado que se trata de una versión beta del comando, las debilidades expuestas en el apartado anterior podrán ser resueltas en el futuro.

Uno de los objetivos en un futuro próximo es enviar a Stata la sintaxis del comando para su revisión y validación, para que pueda ser implementado como comando de usuario, y que esté disponible su descarga para todos los usuarios.

El comando validation, actualmente trabaja únicamente con modelos de regresión logística, pero el objetivo próximo será desarrollar este útil comando para el resto de modelos de regresión más habituales en el contexto biomédico, modelos de regresión de Cox o modelo mixtos.

8. Bibliografía

1. Anderson, K. M., Odell, P. M., Wilson, P. W., & Kannel, W. B. (1991). Cardiovascular disease risk profiles. American heart journal, 121(1), 293-298.
2. Adams F. The genuine works of hippocrates. Vol 17. Sydenham society; 1849.
3. Altman, D. G., & Royston, P. (2000). What do we mean by validating a prognostic model?. Statistics in medicine, 19(4), 453-473.
4. Knottnerus, J. A. (1995). Diagnostic prediction rules: principles, requirements and pitfalls. Primary Care, 22(2), 341-363.
5. Siontis, G. C., Tzoulaki, I., Castaldi, P. J., & Ioannidis, J. P. (2015). External validation of new risk prediction models is infrequent and reveals worse prognostic discrimination. Journal of clinical epidemiology, 68(1), 25-34.
6. Collins, G. S., Reitsma, J. B., Altman, D. G., & Moons, K. G. (2015). Transparent reporting of a multivariable prediction model for individual prognosis or diagnosis (TRIPOD): the TRIPOD statement. BMC medicine, 13(1),1.
7. Moons, K. G., Altman, D. G., Reitsma, J. B., Ioannidis, J. P., Macaskill, P., Steyerberg, E. W., ... & Collins, G. S. (2015). Transparent Reporting of a multivariable prediction model for Individual Prognosis or Diagnosis (TRIPOD): explanation and elaboration. Annals of internal medicine, 162(1), W1-W73.
8. Counsell, C., & Dennis, M. (2001). Systematic review of prognostic models in patients with acute stroke. Cerebrovascular diseases, 12(3), 159-170.
9. Perel, P., Edwards, P., Wentz, R., & Roberts, I. (2006). Systematic review of prognostic models in traumatic brain injury. BMC medical informatics and decision making, 6(1), 38.
10. StataCorp. 2017. Stata Statistical Software: Release 15. College Station, TX: StataCorp LLC
11. R Core Team. 2016. R: A language and environment for statistical computing. R Foundation for Statistical. Computing, Vienna, Austria. URL https://www.R-project.org/.
12. Steyerberg, E. W., Bleeker, S. E., Moll, H. A., Grobbee, D. E., & Moons, K. G. (2003). Internal and external validation of predictive models: a simulation study of bias and precision in small samples. Journal of clinical epidemiology, 56(5), 441-447.

13. Peduzzi, P., Concato, J., Kemper, E., Holford, T. R., & Feinstein, A. R. (1996). A simulation study of the number of events per variable in logistic regression analysis. Journal of clinical epidemiology, 49(12), 1373-1379.
14. Steyerberg, E. W., Eijkemans, M. J., Harrell, F. E., & Habbema, J. D. F. (2000). Prognostic modelling with logistic regression analysis: a comparison of selection and estimation methods in small data sets. Statistics in medicine, 19(8), 1059-1079.
15. Steyerberg, E. W., Harrell Jr, F. E., Borsboom, G. J., Eijkemans, M. J. C., Vergouwe, Y., & Habbema, J. D. F. (2001). Internal validation of predictive models: efficiency of some procedures for logistic regression analysis. Journal of clinical epidemiology, 54(8), 774-781.
16. Harrell FE Jr: Regression modeling strategies. With applications to Linear Models, Logistic Regression, and Survival Analysis. New York, NY: Springer; 2001.
17. Hanley J.A., McNeil B.J. (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology. 143: 29-36.
18. http://www.hrc.es/bioest/Reglog_10.html#biblio3
19. Somers, R. H. (1962). A new asymmetric measure of association for ordinal variables. American sociological review, 799-811.
20. Mittlböck, M., & Schemper, M. (1996). Explained variation for logistic regression. Statistics in medicine, 15(19), 1987-1997.
21. Van Houwelingen, J. C., & Le Cessie, S. (1990). Predictive value of statistical models. Statistics in medicine, 9(11), 1303-1325.
22. Steyerberg, E. W. (2018). Validation in prediction research: the waste by data-splitting. Journal of clinical epidemiology.
23. Steyerberg, E. W., Uno, H., Ioannidis, J. P., Van Calster, B., Ukaegbu, C., Dhingra, T., ... & Kastrinos, F. (2018). Poor performance of clinical prediction models: the harm of commonly applied methods. Journal of clinical epidemiology, 98, 133-143.
24. Steyerberg, E. W., Bleeker, S. E., Moll, H. A., Grobbee, D. E., & Moons, K. G. (2003). Internal and external validation of predictive models: a simulation study of bias and precision in small samples. Journal of clinical epidemiology, 56(5), 441-447.
25. Altman, D. G., Vergouwe, Y., Royston, P., & Moons, K. G. (2009). Prognosis and prognostic research: validating a prognostic model. Bmj, 338, b605.
26. Steyerberg EW, Harrell FE Jr. Prediction models need appropriate internal, internal-external, and external validation. J Clin Epidemiol 2016;69:245e7.

27. Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. In Encyclopedia of database systems (pp. 532-538). Springer, Boston, MA.
28. Efron, B. (1992). Bootstrap methods: another look at the jackknife. In Breakthroughs in statistics (pp. 569-593). Springer, New York, NY.
29. Brunelli, A. (2014). A synopsis of resampling techniques. Journal of thoracic disease, 6(12), 1879
30. Steyerberg, E. W., Borsboom, G. J., van Houwelingen, H. C., Eijkemans, M. J., & Habbema, J. D. F. (2004). Validation and updating of predictive logistic regression models: a study on sample size and shrinkage. Statistics in medicine, 23(16), 2567-2586.
31. Vach, K., Sauerbrei, W., & Schumacher, M. (2001). Variable selection and shrinkage: comparison of some approaches. Statistica Neerlandica, 55(1), 53-75.

9. Anexo

```
/*--------------VALIDACION INTERNA MODELO REGRESION LOGISTICA---------------*/
/*--------------------------------------------------------------------------*/
capture program drop validation
program validation, sortpreserve
        version 15.1
             preserve
             syntax [varlist] [if] [in], reps(integer) [dummy1(string)]
[dummy2(string)] [pr(real .1) pe(real .001)] [seed(integer 1)] [graph]
[group(integer 10)]
quietly {
local sThisProg "validation"
if "`sThisProg'" == "validation" {
       if "`e(cmd)'" != "logit" & "`e(cmd)'" != "logistic" {
             di as err "Error: `sThisProg' must be executed after -logit- or -
logistic-."
             exit 1, clear
             }
       }
if `"`seed'"' != "1" {
       set seed `seed'
       }
if `"`group'"' == "10" {
       local group = 10
       }
if `"`group'"' != "" {
       local group = `group'
       }
local yvar = e(depvar)
local xvars = substr(e(cmdline),length(e(cmd))+length(e(depvar))+2,.)
local dummy =
substr(e(cmdline),strpos(e(cmdline),"(")+1,strpos(e(cmdline),")")-
strpos(e(cmdline),"(")-1)
local xvars2 = subinstr("`xvars'", "`dummy'" , "", .)
local xvars2 = stritrim(subinstr(subinstr("`xvars2'", "(" , "", .), ")" , "",
.))
local comando = e(cmd)
`comando' `yvar' `xvars'
estimates title: The final model
estimates store final
local int bin = round(sqrt(e(N)))
predict p
roctab `yvar' p if e(sample)
local AUC : di %4.3f r(area)
```

```
local AUC1 = r(lb)
local AUC2 = r(ub)
matrix rcoefs = e(b)
estat gof, group(`group')
local HL = r(chi2)
local p_value : di %4.3f r(p)
local df = r(df)
if `"`varlist'"' != "" {
       local xvars2 `xvars2' `varlist'
       }
`comando' `yvar' `xvars2' `dummy' `dummy1' `dummy2'
estimates title: The maximum model
estimates store maximum
local heuristic = (e(chi2)-e(df_m))/e(chi2)
di "heuristic" `"`heuristic'"'
local xvars_maximum =
subinstr(subinstr(substr(e(cmdline),length(e(cmd))+length(e(depvar))+2,.),"("
, "", .),")" , "", .)
tokenize "`xvars_maximum'"
local i = 1
while "`1'" != "" {
    local x`i'="`1'"
       local count`i'=0
    local i=`i'+1
    macro shift
       }
local numx=`i'-1
local numreps=1
tempfile tempds
matrix resultados=J(`reps',7,.)
while `numreps' <= `reps' {
       di "numreps: " `numreps'
       di "reps: "`reps'
       matrix resultados[`numreps',1]=`AUC'
       quietly save `tempds', replace
       bsample
       sw, pr(`pr') pe(`pe'): logistic `yvar' `xvars2' (`dummy') (`dummy1')
(`dummy2')
       lroc, nograph
       matrix resultados[`numreps',2]=r(area)
       estat gof, group(`group')
       matrix resultados[`numreps',5]=r(chi2)
       matrix resultados[`numreps',6]=r(p)
       mat b=e(b)
       local keepx : colnames(b)
       tokenize `keepx'
```

```
di "keepx : " "`keepx'"
local i 1
local xlist ""
while "`1'"~="_cons"  {
       local keepx`i'="`1'"
       local xlist `xlist' `keepx`i''
       local i=`i'+1
       macro shift
}
di "lista : " "`xlist'"
local numkeep=`i'-1
local i 1
while `i'<=`numx'  {
       local j 1
       while `j'<=`numkeep'  {
              if "`x`i''"=="`keepx`j''"  {
                     local count`i'=`count`i''+1
              }
              local j=`j'+1
       }
       local i=`i'+1
}
restore, preserve
tab `yvar'
predict xb, xb
`comando' `yvar' xb, coef
lroc, nograph
matrix resultados[`numreps',3]=r(area)
matrix resultados[`numreps',4]= resultados[`numreps',2]-
resultados[`numreps',3]
matrix slope = e(b)
matrix resultados[`numreps',7]=slope[1,1]
quietly use `tempds', clear
local numreps=`numreps'+1
restore, preserve
}
clear
set obs `reps'
svmat resultados
rename (resultados1-resultados7) (apparent_performance boot_performance
test_performance optimism HL_boot pvalor_boot slope)
sum optimism
local est_optimism = r(mean)
local AUCadj : di %4.3f `AUC'-`est_optimism'
local lower_AUCadj = `AUC1'-`est_optimism'
local upper_AUCadj = `AUC2'-`est_optimism'
```

```
gen sig_boot = pvalor_boot < 0.05, after(pvalor_boot)
ci proportions sig_boot, exact
local sig_boot = r(N)*r(proportion)
local prop_boot = r(proportion)*100
local prop_boot_l = r(lb)*100
local prop_boot_u = r(ub)*100
summ slope
local slope = r(mean)
restore, preserve
if `"`graph'"' == "graph" {
	estimates restore final
	predict p
	roctab `yvar' p, graph ytitle(, size(small)) xtitle(, size(small))
ylabel(, labsize(vsmall)) xlabel(, labsize(vsmall) nogrid) subtitle(Area under
ROC curve, size(medsmall) color(dknavy)) note("AUC apparent = `AUC'"  "AUC
adjusted = `AUCadj'", position(5) ring(0) box bmargin(small) linegap(1)
justification(center) alignment(middle)) name(auc, replace) histogram p if
`yvar' == 0, bin(`bin') fcolor(navy%50) lcolor(navy) xscale(alt) xlabel(,
labels labsize(small)) xtitle(, size (zero)) note(`yvar' No, position(1)
ring(0) box bmargin(small) justification(center) alignment(middle))
plotregion(margin(zero)) name(hist0, replace)
	histogram p if `yvar' == 1, bin(`bin') fcolor(navy) lcolor(dknavy)
xlabel(, labels labsize(small)) yscale(reverse) xtitle(Probabilidad `yvar',
size(medsmall)) note(`yvar' Yes, position(7) ring(0) box bmargin(small)
justification(center) alignment(middle)) plotregion(margin(zero))name(hist1,
replace)
	graph combine hist0 hist1, colfirst xcommon ycommon rows(2) imargin(0 0
0 0) subtitle(Distribution of probabilities, size(medsmall) color(dknavy))
name(prob, replace)
	graph combine auc prob, col(1) title(Discrimination plots) imargin(0 0 2
2) fxsize(50) name(discrimination, replace)
	graph close
	xtile quantil = p, nquantiles(`group')
	margins, over(quantil) noesample
	matrix prediction = r(table)'
	proportion `yvar', over(quantil)
	matrix observed = r(table)'
	clear
	svmat observed
	drop if _n <= e(N_over)
	svmat prediction
	quiet twoway (scatter observed1 prediction1) (scatteri 0 0 1 1,
recast(line)) (pcspike observed5 prediction1 observed6 prediction1),
ytitle(Risk of outcome observed, size(vsmall)) ///
ylabel(0(0.2)1) xtitle(Risk of outcome predicted, size(vsmall)) ylabel(,
labsize(vsmall)) xlabel(0(0.2)1, labsize(vsmall)) fxsize(50) fysize(50)
```

```
title(Calibration plot) subtitle(Observed vs. Predicted probabilities,
size(small) color(dknavy)) legend(off) note(Hosmer Lemeshow test.  p value =
`p_value', size(vsmall) position(5) ring(0) box bmargin(small)
justification(center) alignment(middle)/*fcolor(white)*/) name(calibration,
replace)
      graph close
      graph combine discrimination calibration, col(2) imargin(0 0 0 0)
      }
}
estimates replay maximum
estimates replay final
di _newline
di as text "Apparent performance"
di as text "{hline 55}"
di as text _s(15) "ROC area = " /*"{c |}"*/ as res %7.3f `AUC' _col(37) as
text "95%CI " as res "(" %5.3f `AUC1' "-" %5.3f `AUC2' ")"
di as text "Hosmer-Lemeshow chi2(" as res `df' as text ") = " /*"{c |}"*/ as
res %7.3f `HL' _s(3) as text "Prob > chi2 = " as res %5.3f `p_value'
di as text _s(4) "Heuristic shrinkage = " as res %7.3f `heuristic'
di as text "{hline 55}"
di _newline
di as text "Bootstrap" _s(19) "Number of replications: " as res %3.0f `reps'
di as text "{hline 55}"
di as text _s(15) "Optimism = " as res %7.3f `est_optimism'
di as text _s(4) "ROC area (adjusted) = " as res %7.3f `AUCadj' _col(37) as
text "95%CI " as res "(" %5.3f `lower_AUCadj' "-" %5.3f `upper_AUCadj' ")"
di as text _s(4) "Bootstrap shrinkage = " _s(2) as res %5.3f `slope'
di as text "HL test significatives:"
di as text _s(17) "Number =" _s(2) as res %3.0f `sig_boot'
di as text _s(13) "Proportion =" _s(3) as res %3.1f `prop_boot' "%" _col(37)
as text "95%CI " as res "(" %3.1f `prop_boot_l' "%-" %3.1f `prop_boot_u' "%)"
di as text "{hline 55}"
local i=1
disp _n(1) "Number of replications: " as res %3.0f `reps'
disp as text "Summary: (Number of times each variable is selected)"
di as text "{hline 55}"
while `i'<=`numx' {
    disp _s(3) as text "`x`i'':" _col(25) as res "`count`i''"
    local i=`i'+1
    }
di as text "{hline 55}"
restore
estimates clear
end
```