REGRESIÓN LINEAL CON REMUESTREO

Bonilla A. Enrique, Canales P. Jackeline, Guizado R. José, Vásquez V. Christian

I. INTRODUCCIÓN

Las técnicas de re-muestreo son un conjunto de métodos desarrollados para calcular y estimar estadísticos, basándose en técnicas computacionales que evitan los cálculos complejos de la teoría estadística clásica. Estas técnicas buscan resolver uno de los problemas fundamentales en estadística, evaluar la variabilidad de una estimación (error estándar) en particular. Actualmente, con la disponibilidad de poderosos recursos informáticos, es posible realizar múltiples cálculos de muestra del conjunto de datos original según sea necesario, lo que nos permite aplicar técnicas de re-muestreo de una manera accesible.

El proceso de generar un gran número de muestras de un conjunto de datos inicial y realizar cálculos posteriores para inferir distribuciones muestrales de un estimado se le conoce como re-muestreo. Existen múltiples técnicas de esta naturaleza, de las cuales conoceremos más a detalle en este informe (bootstrap y validación cruzada con sus variantes)

Para los modelos de regresión lineal, donde los supuestos de distribuciones son de gran importancia, el re-muestreo nos da la alternativa o ventaja de no necesitar suposiciones sobre la distribución de la información, así como la implementación flexible con mínimo esfuerzo matemático.

Efron, B. (1979) propuso el método bootstrap de re-muestreo para aproximar la distribución en el muestreo de un estadístico, en específico, detalló el uso para aproximar el sesgo y varianza, así como la construcción de intervalos de confianza para un estadístico.

Es de amplio uso para la estimación de cualquier estadístico, así como la aplicación para regresión lineal con los métodos de observaciones y de residuales. (Efron y Tibshirani. 1993).

Por otro lado, García et al (2015) indican que para evitar problemas de sobreajuste y controlar el rendimiento de los modelos, las técnicas de validación cruzada pueden ser muy efectivas. Mckinney (2018) también resalta la importancia de dicha técnica, resaltando que el uso óptimo de validación cruzada nos permite tener unos mejores niveles de predicción y robustez ante nueva información a evaluar para el modelo.

La denominación Jackknife (navaja) alude a lo multiuso y de gran utilidad, es decir, describe su versatilidad a pesar de que los problemas específicos pueden ser más eficientemente resueltos con una herramienta diseñada para tal fin (Cameron y Trivedi, 2005).

El método fue desarrollado por Quenouille en 1949 con la intención de reducir el sesgo de un estimador, y fue bautizado con este nombre por Tukey 1958 que lo generalizó como método general de estimación.

Quenouille, M (1949) inventó este método con la intención de reducir el sesgo de la estimación de la muestra. Tukey amplió este método para ello supuso que, si las réplicas pudieran considerarse distribuidas de manera idéntica e independiente, entonces podría hacerse una estimación de la varianza del parámetro de muestra y que se distribuiría aproximadamente como una variable t con n -1 grados de libertad (donde n es el tamaño de la muestra) (Efron, 1982).

Finalmente, tenemos como objetivo general Evaluar métodos de pre-procesamiento en modelos de regresión lineal y su efecto en la mejoría del ajuste y solución de posibles problemas de sobrejauste o subajuste. Las técnicas de re-muestreo para regresión lineal evaluadas fueron bootstrap, k-fold cross validation, repeated k-fold cross validation y leave one out cross validation (LOOCV) con una información de prueba, para la creación de un modelo de predicción de los kilos cosechados semanales de bayas por planta en el cultivo de arándano según variables exógenas del clima y del número de bayas presentes. Como objetivos específicos, tenemos: 1) Calcular los estimadores y errores estándar de los coeficientes de regresión lineal al aplicar las técnicas de re-muestreo, y 2) Identificar la técnica de re-muestreo con mayor ajuste y/o menor error de estimación para el modelo predictivo con regresión lineal.

II. MARCO TEÓRICO

Los métodos y técnicas de regresión estudian la elaboración de modelos con el fin de explicar la dependencia entre una variable respuesta o dependiente y la(s) variable(s) explicativa(s) o dependiente(s); si hablamos en específico del modelo de regresión lineal, tiene lugar cuando la dependencia es de tipo lineal.

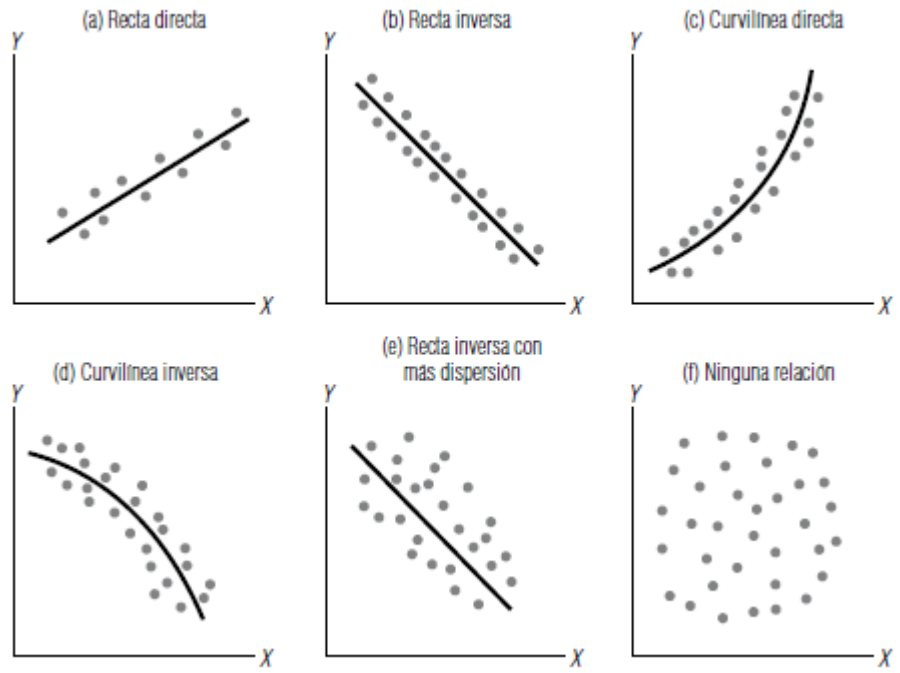
2.1 Regresión lineal simple

En el análisis de regresión simple intervienen una variable dependiente (Y) y una independiente (X) y en la cual la relación entre ellas se aproxima por medio de una línea recta.

En principio no sabemos si las variables a analizar están relacionadas o no, o si en caso de haber dependencia es significativa o no. Para esto, el gráfico de dispersión nos permite identificar si existe menor o mayor grado de asociación entre las variables.

Figura 1.

Relaciones entre dos variables (X e Y) según gráfico de dispersión.



Nota. Recuperado de "*Estadística para administración y economía*", por Levin, R. y Rubin, D., 2010.

La función del modelo de regresión lineal es:

$$Y = \beta_0 + \beta_1 X + \varepsilon \quad (1)$$

Se observa que:

- La media de Y, para un valor fijo X, varía linealmente con X.
- El parámetro β_0 es la ordenada al origen del modelo (intercepto), valor que toma la media de Y cuando X es inexistente, y β_1 , la pendiente, que puede interpretarse como el incremento de la variable dependiente por cada incremento en una unidad de la variable independiente. Estos parámetros son inicialmente desconocidos.
- Bajo los supuestos de la función, podemos ver que los factores o causas que influyen en la variable respuesta Y son primordialmente ocasionados por una variable explicativa (X) y por un conjunto de factores no controlados bajo el nombre de error aleatorio (ε), que en conjunto provocan que la dependencia entre las variables dependiente e independiente no sea perfecta, sino que esté sujeta a incertidumbre.

Dentro de las ventajas del modelo de regresión lineal, tenemos lo siguiente:

- Sencillez en su aplicación y entendimiento.
- Las ecuaciones lineales son de fácil interpretación, por ende, es particularmente útil cuando la relación a modelar no es extremadamente compleja y/o no tiene mucha información.
- Es menos propenso al sobreajuste.

Por otro lado, las desventajas de la técnica son:

- Sólo se aplica cuando existe relación lineal entre la variable dependiente e independiente, limitando su uso a múltiples escenarios.
- Es sensible a valores atípicos.
- Limitado al cumplimiento de supuestos previos.

Métodos de estimación

Dada una función de regresión muestral $\hat{Y}_i = \hat{\beta}_1 + \hat{\beta}_2 X_i + \hat{u}_i$, existen los siguientes métodos de estimación de parámetros:

a) Mínimos cuadrados ordinarios

Tiene como objetivo identificar los valores de $\hat{\beta}_1$ y $\hat{\beta}_2$ que permitan el mejor ajuste posible del modelo, minimizando el error (residuos).

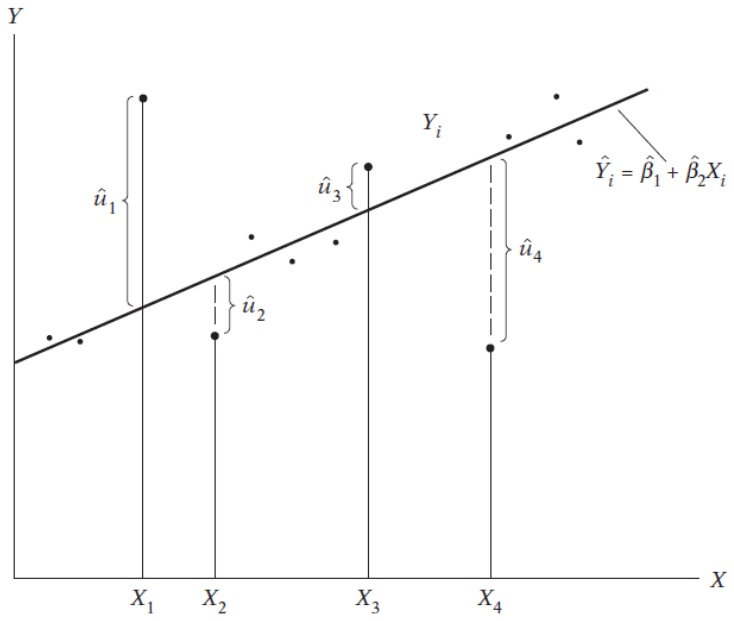
Dada la ecuación de los residuos:

$$\widehat{u_\iota} = Y_i - \widehat{Y_i}$$

$$\widehat{u_\iota} = Y_i - \hat{\beta}_1 - \hat{\beta}_2 X_i$$

Figura 2.

Relaciones entre dos variables (X e Y) según gráfico de dispersión



Nota. Recuperado de "*Econometría*", por Gujarati, D. y Porter, D., 2010.

Al minimizar el total de los errores : $\sum \widehat{u^2}_\iota$

Obteniendo los coeficientes:

$$\hat{\beta}_2 = \frac{n\sum X_i Y_i - \sum X_i \sum Y_i}{n\sum X_i^2 - \left(\sum X_i\right)^2}$$

$$= \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sum (X_i - \bar{X})^2}$$

$$= \frac{\sum x_i y_i}{\sum x_i^2}$$

$$\hat{\beta}_1 = \frac{\sum X_i^2 \sum Y_i - \sum X_i \sum X_i Y_i}{n\sum X_i^2 - \left(\sum X_i\right)^2}$$

$$= \bar{Y} - \hat{\beta}_2 \bar{X}$$

b) Máxima verosimilitud

Permite estimar los parámetros desconocidos de manera que la probabilidad de observar las Y dadas sea lo más alta (o máxima) posible. Por consiguiente, se tiene que encontrar el máximo de la función en la ecuación de verosimilitud:

$$\mathrm{FV}(\beta_1, \beta_2, \sigma^2) = \frac{1}{\sigma^n\left(\sqrt{2\pi}\right)^n} \exp\left\{-\frac{1}{2}\sum \frac{(Y_i - \beta_1 - \beta_2 X_i)^2}{\sigma^2}\right\} \quad (2)$$

Donde $\beta_1 \, y \, \beta_2$ son los parámetros y σ^2 es la varianza. Los estimadores de MV y MCO de $\beta_1 \, y \, \beta_2$ son idénticos, sin embargo, los estimadores de la varianza no suelen ser en la misma medida idénticos. (Gujarati y Porter, 2010)

Asimismo, se deben respetar los siguientes supuestos:

- *Normalidad de los errores:* Para cada valor de la variable X, los residuos $e_i = (\hat{Y}_i - Y_i)$ tienen distribución normal.
- *Homogeneidad de varianzas:* Para cada valor de la variable X, la varianza de los residuos e_i debe ser la misma.
- *Independencia de errores:* Los residuos deben ser independientes entre sí.
- *Linealidad:* Si se presenta una relación lineal significativa entre las variables.

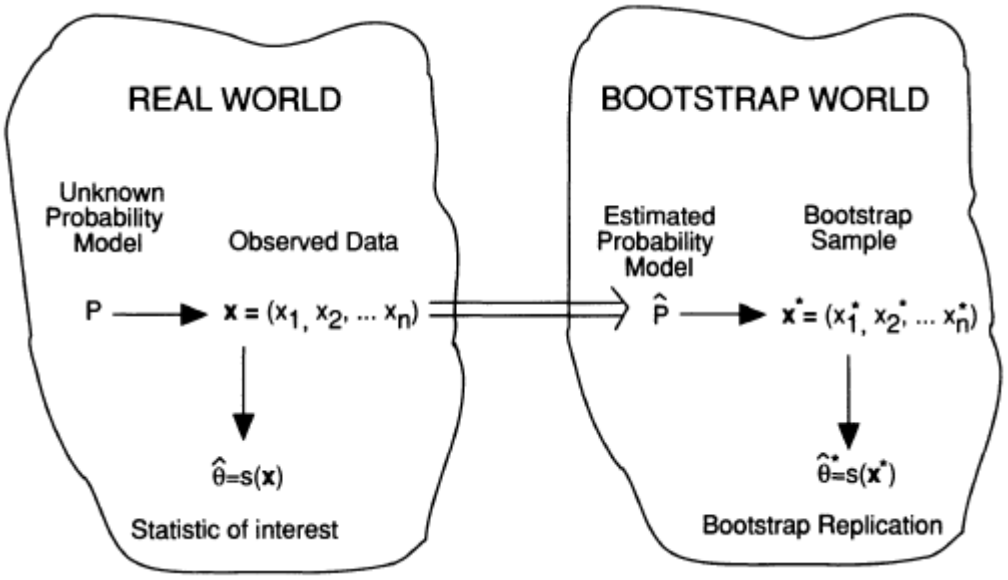
2.2 Bootstrap

Es un método de re-muestreo inicialmente propuesto por Bradley Efron (1979). Se utiliza para aproximar la distribución en el muestreo de un estadístico y se usa frecuentemente para aproximar el sesgo o la varianza de un análisis estadístico, así como para construir intervalos de confianza o realizar contrastes de hipótesis sobre parámetros de interés.

El objetivo del bootstrapping es inferir sobre una población a partir de datos de muestra, esta puede ser obtenida mediante un nuevo muestreo de los datos de la muestra y realizando la inferencia sobre una muestra a partir de datos remuestreados. La ventaja principal del bootstrap es que no requiere de hipótesis sobre el mecanismo generador de datos, ni suposiciones de distribuciones teóricas, todo esto mediante el uso de recursos computacionales que faciliten los procedimientos.

Figura 3

Diagrama esquemático de la aplicación de Bootstrap a una muestra para estimación de probabilidad.



Nota. Recuperado de "Introduction to the Bootstrap", por Efron, B. y Tibshirani, R., 1993.

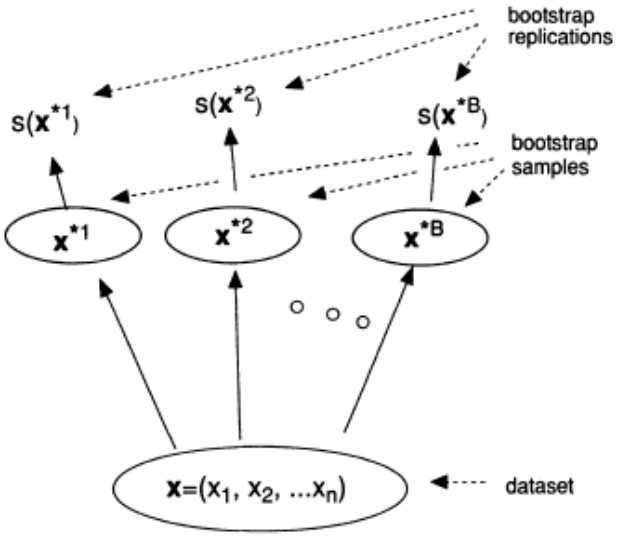
Muestra bootstrap

En términos aplicativos, podemos definir la muestra bootstrap de la siguiente forma:

Dada la muestra aleatoria: $x = (X_1, X_2, \dots, X_n)$ de tamaño "*n*", una muestra bootstrap es la representada : $x^* = (X^*_1, X^*_2, \dots, X^*_n)$, del mismo tamaño "*n"* y escogida con reemplazo. De tal forma, al tener muestras con reemplazo y del mismo tamaño, tenemos muestras con función de distribución empírica F_n.

Figura 4.

Esquema del proceso bootstrap para la generación de muestras



Nota. Recuperado de "*Introduction to the Bootstrap*", por Efron, B. y Tibshirani, R., 1993.

Estimador bootstrap del error estándar de un estimador

Sea θ el valor del estimador en la muestra bootstrap x^*, el error estándar bootstrap del estadístico θ estará dado por:

$$\widehat{ee}(\hat{\theta}) = ee_{\hat{F}}(\hat{\theta}^*) \tag{3}$$

El estimado bootstrap del error estándar, es el error estándar de un conjunto de muestras bootstrap elegidas de la muestra original. Por ende, si se toman *B* muestras bootstrap, el error estándar el estimado es:

$$\widehat{ee}_B(\hat{\theta}) = \sqrt{\frac{\sum_{i=1}^{B}(\hat{\theta}_i^{\,*} - \bar{\hat{\theta}}^*)^2}{B-1}} \tag{4}$$

Donde: $\bar{\theta}^* = \sum_{i=1}^{B} \frac{\theta_i^{\,*}}{B}$

Bootstrap en regresión lineal

En el modelo clásico de regresión lineal se tiene un conjunto de *n* parejas de observaciones $x_1, x_2, \dots, x_n$, donde cada X_i es un par de datos: $x = (c_i, y_i)$.

Efron y Tibshirani (1993) definen cada c_i como un vector de dimensión *1 x p* tal que $c_i = (c_{i1}, c_{i2}, \dots, c_{ip})$, y se suele denominar como vector de covariables o predictores. Por otro lado, y_i es un número real referente a la variable respuesta.

Se define la esperanza condicional de la respuesta y_i dado el predictor c_i como $\mu_i = (y_i|c_i)$, para *i = 1,2, … , n.*

La suposición inicial de los modelos lineales es que μ_i es una combinación lineal de los componentes del vector c_i , expresado como:

$$\mu_i = c_i\beta = \sum_{j=1}^{p} c_{ij}\beta_j \qquad (5)$$

De tal forma que los parámetros (vector de parámetros) $\beta = (\beta_1, \beta_1, \dots, \beta_p)^T$ son desconocidos y se buscan estimarlos mediante las observaciones $x_1, x_2, \dots, x_n$.

La estructura habitual para las relaciones de covariables y respuesta es:

$$y_i = c_i\beta + \varepsilon_i \qquad (6)$$

Para la estimación de parámetros de la regresión (β), tomando de partida un valor hipotético "*b*" de β, tenemos el error cuadrático medio (ECM):

$$ECM(\boldsymbol{b}) = \sum_{i=1}^{n}(y_i - c_i\boldsymbol{b})^2 \qquad (7)$$

Y el estimador de mínimos cuadrados ordinarios de β es el valor $\hat{\beta}$, del valor "b" que minimiza el ECM:

$$ECM(\hat{\beta}) = \min_{b}(ECM(\boldsymbol{b})) \qquad (8)$$

Si **C** (llamada matriz de diseño), de orden *n* x *p*, donde la fila i-ésima es $\boldsymbol{c_i}$ y se denomina $\boldsymbol{y}$ al vector $(y_1, y_2, \dots, y_n)^T$; entonces el estimador de mínimos cuadrados ordinarios mediante ecuaciones es: $\mathbf{C}^T\mathbf{C}\hat{\beta} = \mathbf{C}^T\boldsymbol{y}$

Despejando : $\hat{\beta} = (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\boldsymbol{y}$

El problema principal en la regresión es tratar de estimar el vector de parámetros β usando los datos. Tal como se describe anteriormente, la aplicación de bootstrap en regresión se centra en la inferencia sobre el vector de parámetros β , en la estimación del error de predicción $\widehat{ee}(\hat{\theta})$ y en la selección de variables.

Existen dos maneras de aplicar bootstrap en regresión: bootstrap de observaciones y bootstrap de residuales.

Bootstrap de observaciones

El método de observaciones o de pares de valores se basa en la aplicación bootstrap en regresión, re-muestreando parejas de valores $x_i = (c_{i\,,} y_i)$.

Una muestra bootstrap siguiendo dicho método sería:

$x^* = \{(c_{i1\,,} y_{i1}), (c_{i2\,,} y_{i2}),\ldots, (c_{in\,,} y_{in})\}$

Para $i_{1\,,} i_2, \ldots, i_n$, que es una muestra aleatoria de números enteros entre 1 y *n.*

Posteriormente, se aplica el modelo de regresión obteniendo una secuencia de estimados bootstrap β^*.

Bootstrap de residuales

Complementando lo anterior, al ser β desconocido, definimos los errores aproximados o residuos como (usando $\hat{\beta}$): $\hat{\varepsilon}_i = y_i - c_i\hat{\beta}$, para *i* = 1, 2, … , *n*.

Según Efron y Tibshirani (1993), el modelo de probabilidad $P = (\beta, F)$ tiene 2 componentes, donde β es el vector de parámetros de la regresión y *F* es la distribución de los errores. Ambos valores se estimarán $\hat{\beta}$, $\hat{F}$.

A partir de la estimación de los valores $\hat{P} = (\hat{\beta}, \hat{F})$, es que se calculan las muestras bootstrap $\hat{P} \rightarrow x^*$. Para este cálculo, primero tomamos una muestra aleatoria de los términos de error:

$\hat{F} \rightarrow (\varepsilon^*{}_1, \varepsilon^*{}_2, \ldots, \varepsilon^*{}_n) = \varepsilon^*$

Donde cada $\varepsilon^*{}_i$ es igual a cualquiera de los n valores de $\hat{\varepsilon}_j$ con probabilidad 1/*n*. De tal modo, las nuevas respuestas (variable dependiente) bootstrap se generan mediante:

$$y^*{}_i = c_i\hat{\beta} + \varepsilon^*{}_i$$

Para *i*=1, 2, … , *n*; donde $\hat{\beta}$ es el mismo para todo *i.*

Finalmente, las muestras bootstrap serían:

$$x^*{}_i = (c_{i\,,} y^*{}_i)$$

Las cuales generan un dataset bootstrap de forma:

$x^* = \{(c_{1,}c_1\hat{\beta} + \hat{\varepsilon}_{i1}), (c_{2,}c_2\hat{\beta} + \hat{\varepsilon}_{i2}),\ldots, (c_{n,}c_n\hat{\beta} + \hat{\varepsilon}_{in})\}$

Ventajas y desventajas al aplicar bootstrap de observaciones y de residuales

Para evaluar cuál método es el más efectivo, es importante cómo se defina el modelo de regresión. Si en el modelo se asume que los errores $(y_i - c_i\beta)$ no dependen de c_i , entonces implica que tiene la misma distribución *F* sin importar dichos valores de c_i .

Ante lo dicho anteriormente, el bootstrap de observaciones lo único que se requiere es que las parejas de valores originales $x_i = (c_{i,}y_i)$ se re-muestreen de manera aleatoria.

Con esto podemos concluir que el método de observaciones es menos sensible a los supuestos (normalidad, varianza constante e independencia de residuales) ya que la única suposición detrás de dicho método sería que los pares originales se re-muestreen bajo una distribución *F*, donde *F* es una distribución en (p + 1) vectores dimensionales (c, y). (Efron y Tibshirani, 1993)

Sin embargo, este método tiende a producir distribuciones con mayor variabilidad. Y, por otro lado, el bootstrap de residuales considera a las variables predictoras como fijas no aleatorias.

Adicionalmente, es importante resaltar que ambos métodos son sensibles a la multicolinealidad de las variables predictoras.

Estimación de intervalos de confianza

Asumiendo que los errores del modelo de regresión lineal se distribuyen de forma normal, con media cero y varianza constante, y que por otro lado cumpla con el supuesto de independencia, se puede establecer que un intervalo de confianza del $(1 - \alpha)$ % para el coeficiente de regresión poblacional β_j :

$$IC(\beta_j) = \left[\widehat{\beta_j} - t_{(\frac{\alpha}{2},n-p-1)}\, ee(\widehat{\beta_j})\,;\, \widehat{\beta_j} + t_{(1-\frac{\alpha}{2},n-p-1)}\, ee(\widehat{\beta_j})\right]$$

- **<u>Método estándar:</u>** La diferencia aquí es que el error estándar clásico se sustituye por el error estándar bootstrap.

$$IC(\beta_j) = \left[\widehat{\beta_J} - t_{(\frac{\alpha}{2},n-p-1)}\, ee(\widehat{\beta_J}^*)\,;\, \widehat{\beta_J} + t_{(1-\frac{\alpha}{2},n-p-1)}\, ee(\widehat{\beta_J}^*)\right]$$

- **<u>Método de percentiles:</u>** En este método se encuentra los percentiles del $(\alpha/2)$ % y $(1 - \alpha/2)$ % de la distribución empírica acumulada de los estimados bootstrap $\widehat{\beta_J}$. El intervalo de confianza sería de la siguiente forma:

$$IC(\beta_j) = \left[F_{\frac{\alpha}{2}}^{-1}(\widehat{\beta_J}^*)\,;\, F_{1-\frac{\alpha}{2}}^{-1}(\widehat{\beta_J}^*)\right]$$

2.3 K – fold Cross Validation (validación cruzada de k pliegues)

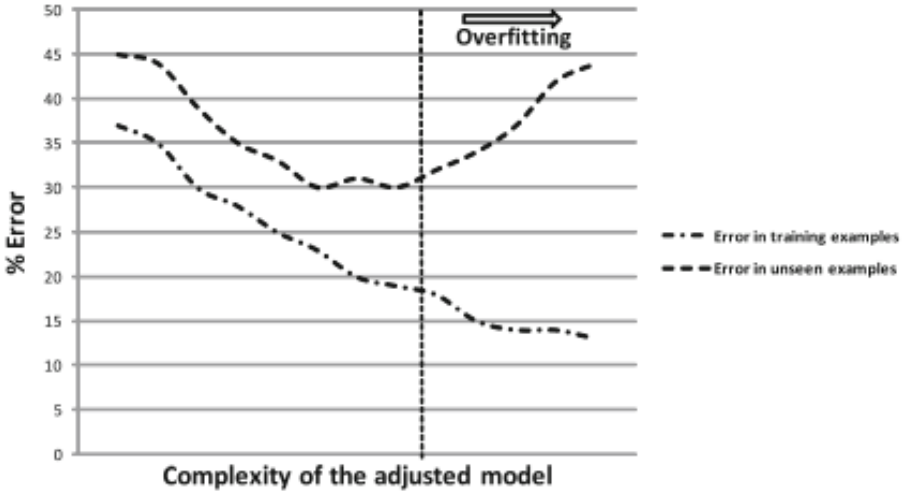
Conceptos básicos o generales

Al utilizar la totalidad de los datos podemos ser conscientes de los problemas de subajuste debidos a un bajo rendimiento del modelo. Ajustar dicho modelo para que se adapte mejor a los datos puede llevar al sobreajuste pero la falta de casos no vistos hace imposible notar esta situación.

Tenga en cuenta también que llevar este procedimiento al extremo puede llevar a un sobreajuste como representado en la Figura 1. Según el razonamiento de la Navaja de Occam, dados dos modelos de errores de generalización similares, uno debería preferir el modelo más simple sobre el más más complejo.

Figura 5.

Evolución típica del porcentaje de error cuando se ajusta un modelo supervisado.



. El subajuste es perceptible en el lado izquierdo de la figura.

El sobreajuste también puede aparecer debido a otras razones como el ruido, ya que puede obligar al modelo a ajustarse erróneamente a regiones falsas del espacio del problema. La falta de datos también provocará un subajuste, ya que las medidas internas que sigue el algoritmo ML sólo pueden tener en cuenta los ejemplos conocidos y su distribución en el espacio.

Para controlar el rendimiento del modelo, evitar el sobreajuste y tener una estimación generalizable de la calidad del modelo obtenido, se han introducido varios esquemas de partición en la literatura. El más común es la Validación Cruzada k-Fold (k-FCV) (García, Luengo y Herrera, 2015).

La validación cruzada es similar al método de submuestreo aleatorio repetido, pero el muestreo se realiza de forma que no se solapen dos conjuntos de pruebas. En la validación cruzada de k pliegues, el conjunto de aprendizaje disponible se divide en subconjuntos separados de un tamaño aproximadamente igual. Aquí, "pliegue" se refiere al número de subconjuntos resultantes. Esta partición se realiza mediante el muestreo aleatorio de casos del conjunto de aprendizaje sin reemplazo. El modelo se entrena utilizando k-1 subconjuntos que, en conjunto, representan el conjunto de entrenamiento. A continuación, el modelo se aplica al subconjunto restante, que se denomina conjunto de validación, y se mide el rendimiento. Este procedimiento se repite hasta que cada uno de los k subconjuntos haya servido de conjunto de validación. La media de las k mediciones de rendimiento en los k conjuntos de validación es el rendimiento de validación cruzada. La figura 1 ilustra este proceso para k= 10, es decir, una validación cruzada de 10 veces. En el primer pliegue, el primer subconjunto sirve como conjunto de validación $D_{val,1}$ y los nueve subconjuntos restantes sirven como conjunto de entrenamiento $D_{train,1}$. En el segundo pliegue, el segundo subconjunto es el conjunto de validación y los subconjuntos restantes son el conjunto de entrenamiento, y así sucesivamente (García, Luengo y Herrera, 2015).

La selección de características suele ser una parte integral del proceso de construcción del modelo. En este caso, es crucial que las características predictivas se seleccionen utilizando sólo el conjunto de entrenamiento, y no todo el conjunto de aprendizaje; de lo contrario, la estimación del error de predicción puede estar muy sesgada. Supongamos que las características predictivas se seleccionan basándose en el conjunto de aprendizaje completo, y luego el conjunto de aprendizaje se divide en conjuntos de validación y conjuntos de entrenamiento, lo que significa que la información de los conjuntos de validación se utilizó para la selección de las características predictivas. Pero los datos de los conjuntos de validación sólo sirven para evaluar el modelo; no se nos permite utilizar estos

datos de ninguna otra manera; de lo contrario, la fuga de información causaría un sesgo a la baja de la estimación, lo que significa que subestima el verdadero error de predicción (Berrar, 2018)..

La validación cruzada se utiliza con frecuencia para ajustar los parámetros del modelo, por ejemplo, el número óptimo de vecinos más cercanos en un clasificador de k-próximos. En este caso, la validación cruzada se aplica varias veces para diferentes valores del parámetro de ajuste, y el parámetro que minimiza el error de validación cruzada se utiliza para construir el modelo final. De este modo, la validación cruzada aborda el problema del sobreajuste (Berrar, 2018).

Definición del modelo

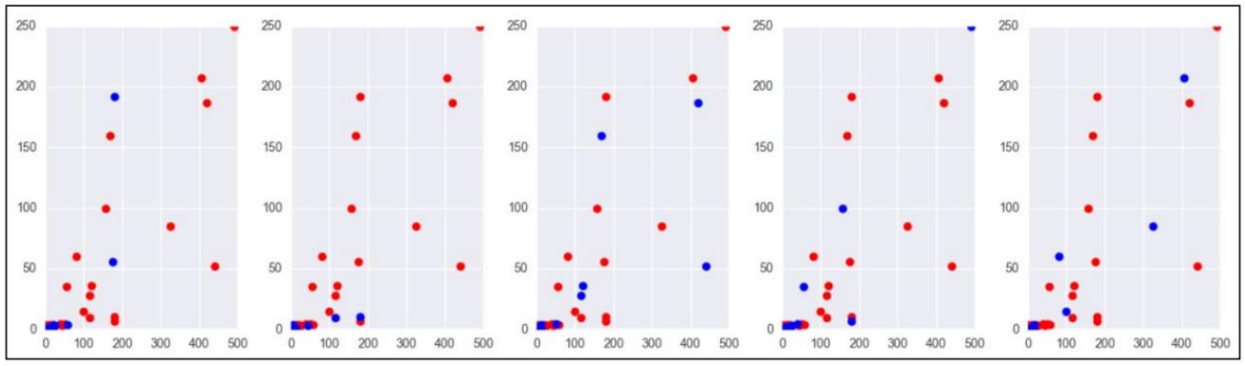
Según García et al, (2015), así es como funciona:

1. Tomaremos un número finito de cortes iguales de nuestros datos (normalmente 3, 5 o 10). Supongamos que este número se llama k.
2. Para cada "pliegue" de la validación cruzada, trataremos k-1 de las secciones como el conjunto de entrenamiento, y la sección restante como nuestro conjunto de prueba.
3. Para los pliegues restantes, se considera una disposición diferente de k-1 secciones para nuestro conjunto de entrenamiento y una sección diferente es nuestro conjunto de entrenamiento.
4. Calculamos una métrica de conjunto para cada pliegue de la validación cruzada.
5. Al final, hacemos una media de nuestras puntuaciones.

La validación cruzada es el uso efectivo de múltiples divisiones de entrenamiento-prueba que se realizan en los mismos datos. Esto se hace por varias razones, pero principalmente porque la validación cruzada es la estimación más honesta del error de nuestro modelo fuera de la muestra.

Figura 6.

Estructura visual de los folios según la partición en prueba y entrenamiento.



Aquí, cada gráfico muestra exactamente la misma población de mamíferos, pero los puntos están rojo si pertenecen al conjunto de entrenamiento de ese pliegue y azul si pertenecen al conjunto del conjunto de pruebas. De este modo, obtenemos cinco instancias diferentes del mismo de aprendizaje automático para ver si el rendimiento es consistente en todos los pliegues.

Si observamos los puntos durante mucho tiempo, nos daremos cuenta de que cada punto aparece en un conjunto de entrenamiento exactamente cuatro veces (k - 1). de entrenamiento exactamente cuatro veces (k - 1), mientras que el mismo punto aparece en un conjunto de prueba exactamente una vez y sólo una vez.

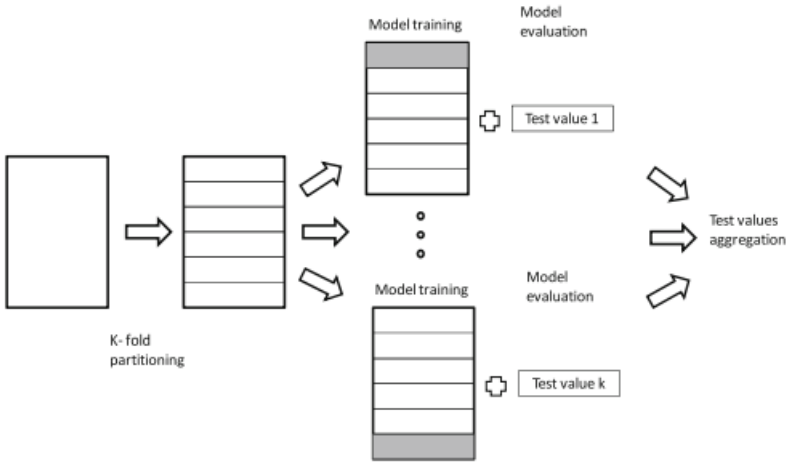
Según García et al (2015), mencionan que algunas consideraciones del modelo son:

1. En k-FCV, los datos originales se dividen aleatoriamente en k pliegues o particiones de igual tamaño.
2. De las k particiones, una se retiene como datos de validación para probar el modelo, y las restantes k - 1 submuestras se utilizan para construir el modelo.
3. Como tenemos k particiones, el proceso se repite k veces y cada una de las k submuestras se utiliza exactamente una vez como datos de validación.

Por último, hay que combinar los k resultados obtenidos de cada una de las particiones de prueba normalmente haciendo una media, para obtener un único valor, como se muestra en la Figura 2. Este procedimiento es muy utilizado, ya que se ha demostrado que estos esquemas asintóticamente convergen a un valor estable, lo que permite realizar comparaciones realistas entre clasificadores.

Figura 7.

Proceso de K – Fold.



El valor de k puede variar, siendo 5 y 10 los más comunes. Este valor debe ajustarse para evitar generar una pequeña partición de prueba mal poblada de ejemplos que pueda sesgar las medidas de rendimiento utilizadas. Si se utilizan conjuntos de datos grandes, se suele utilizar 10-FCV, mientras que para conjuntos de datos más pequeños es más frecuente 5-FCV.

El k-FCV simple también puede llevar a desordenar la proporción de ejemplos de cada clase en la partición de prueba. El método más empleado en la literatura para evitar este problema es el k-FCV estratificado. Coloca un número igual de muestras de cada clase en cada partición para mantener las distribuciones de clase iguales en todas las particiones.

Otros esquemas de validación populares son:

- En 5×2 CV el dataset se divide aleatoriamente en dos subconjuntos A y B. A continuación, el modelo se construye primero con A y se valida con B, y luego el proceso se invierte con el modelo construido con B y validado con A. Este proceso de partición Este proceso de partición se repite como se desee, agregando la medida de rendimiento en cada paso. La Figura 3 ilustra el proceso. La validación cruzada estratificada 5×2 es la variación más utilizada en este esquema.
- Dejar uno fuera es un caso extremo de k-FCV, donde k es igual al número de ejemplos envlos datos. En cada paso sólo se utiliza una instancia para probar el modelo, mientras que el resto de las instancias se utilizan para aprenderlo.

La forma de dividir los datos es una cuestión clave, ya que influirá en gran medida en el rendimiento de los métodos y en las conclusiones que se extraigan a partir de ese momento. Si se realiza realizar una mala partición conducirá seguramente a datos de comportamiento incompletos y/o sesgados sobre del modelo que se está evaluando. Esta cuestión se está investigando activamente en la actualidad, con atención especial al desplazamiento del dataset como un factor decisivo que impone grandes k en k-FCV para alcanzar la estabilidad del rendimiento en el modelo que se está evaluando (García, Luengo y Herrera, 2015).

Métodos de estimación

Según Kohavi, R. (2001), en la validación cruzada de k pliegues, a veces llamada estimación por rotación, los datos D se dividen aleatoriamente en k subconjuntos mutuamente excluyentes (los pliegues) D1; D2; ... ; Dk de tamaño aproximadamente igual. El inductor se entrena y se prueba k veces; cada vez t 2 {1; 2; ...; k}, se entrena con D / Dt y se prueba con Dt. La estimación de la precisión en la validación cruzada es el número total de clasificaciones correctas, dividido por el

número de instancias original. Formalmente, dejemos que D(i) sea el conjunto de pruebas que incluye la instancia xi = <vi; yi>, entonces la estimación de precisión por validación cruzada es:

$$\mathtt{acc}_{\mathrm{CV}} = \frac{1}{n} \sum_{\langle v_i, y_i \rangle \in \mathcal{D}} \delta\big(\mathcal{I}\big(\mathcal{D} \setminus \mathcal{D}_{(i)}, v_i\big), y_i\big) \ . \qquad (4)$$

La estimación de la validación cruzada es un número aleatorio que depende de la división en pliegues. La validación cruzada completa es la media de todas las $\binom{m}{m/k}$ posibilidades de elegir m=k instancias de entre m, pero suele ser demasiado cara. Kohavi, R. (2001), también comenta que excepto en el caso de la validación cruzada de uno en uno (n-fold cross-validation) o leave-one-one (n-fold cross-validation), que siempre es completa, la validación cruzada k-fold es la estimación de la validación cruzada k-fold completa utilizando una única división de los datos en los pliegues. La repetición de la validación cruzada varias veces utilizando diferentes divisiones en pliegues proporciona una mejor estimación de Monte-Carlo a la validación cruzada completa con un coste añadido. En la validación cruzada estratificada, los pliegues se estratifican para que contengan aproximadamente las mismas proporciones de etiquetas que la información original.
Un inductor es estable para un conjunto de observaciones dado y un conjunto de perturbaciones, si induce clasificadores que hacen las mismas predicciones cuando se le dan los conjuntos de datos perturbados.

Características del modelo (resaltar ventajas y desventajas)

La validación cruzada K-fold (KFCV) también llamada validación cruzada V-fold es computacionalmente menos costosa en comparación con LOOCV y LPOCV. Fue introducido por Geisser. El dataset completo de se divide en K subconjuntos de tamaño aproximadamente igual n/K. Cada subconjunto se utiliza sucesivamente como conjunto de validación, mientras que los otros subconjuntos se utilizan para entrenar la superficie de respuesta. El número de ejecuciones de CV es igual a K el número de pliegues que se denota entre 2, …, n pero no debería ser superior a n/3 en un sentido práctico. Si K fuera igual a n, el procedimiento sería idéntico al de LOOCV. El método descrito es un procedimiento de división parcial de los datos, ya que no se utilizan todas las combinaciones posibles de subconjuntos de validación (Beschorner, Voigt, & Vogeler, 2014).

La validación cruzada de K pliegues es un estimador mucho mejor del rendimiento de nuestro modelo, incluso más que nuestra división entrenamiento-prueba (Salvador, Luengo y Herrera, 2015).

Los diagramas muestran claramente que la validación cruzada de k pliegues tiene un sesgo pesimista, especialmente para dos y cinco pliegues. Para las curvas de aprendizaje que tienen una gran derivada en el punto de medición, el pesimismo en la validación cruzada k-fold para k's pequeños es evidente. La mayoría de las estimaciones son razonablemente buenas a 10 pliegues y a 20 pliegues son casi insesgadas (Kohavi, 2001).

Según Ozdemir (2016), algunas de las características de la validación cruzada K-fold son las siguientes

- Es una estimación más precisa del error de predicción OOS que una única de entrenamiento y prueba porque se toman varias divisiones independientes de entrenamiento y prueba y se promedian los resultados juntos. y promediando los resultados juntos.
- Es un uso más eficiente de los datos que las divisiones individuales de entrenamiento-prueba, ya que todo el conjunto de datos se utiliza para múltiples pruebas. de datos se utiliza para varias divisiones de entrenamiento-prueba en lugar de una sola.
- Cada registro se utiliza tanto para el entrenamiento como para la prueba.
- Este método presenta un claro equilibrio entre la eficiencia y el gasto computacional. Un CV de 10 veces es 10 veces más caro computacionalmente que una única de entrenamiento/prueba. (Desventaja)
- Este método puede utilizarse para el ajuste de parámetros y la selección de modelos.

Básicamente, siempre que queramos probar un modelo en nuestra información, ya sea que acabemos de parámetros o de la ingeniería de características, una validación cruzada k-fold es una excelente manera de estimar el rendimiento de nuestro modelo (Ozdemir, 2016).

Los métodos prácticos suelen basarse en procedimientos estadísticos conocidos y fiables, cuyas hipótesis subyacentes, sin embargo, no pueden satisfacerse siempre o sólo son válidas asintóticamente. Los métodos prácticos suelen dividir la información disponible en dos subconjuntos independientes: uno se utiliza para crear un modelo (conjunto de entrenamiento), mientras que el otro se utiliza para calcular la estimación del error de generalización (conjunto de retención).

Las técnicas prácticas más utilizadas para la selección de modelos son: la validación cruzada k-Fold (KCV), el Leave One Out (LOO) y el Bootstrap (BTS). La técnica KCV consiste en dividir la información en k subconjuntos independientes; todos menos uno se utilizan para entrenar un clasificador, mientras que el restante se utiliza como conjunto de retención para evaluar un error de generalización medio. La técnica LOO es análoga a una KCV en la que el número de pliegues es igual al número de patrones disponibles: una muestra se utiliza como hold-out, mientras que las restantes se utilizan para entrenar un modelo. El método BTS, en cambio, es una técnica de remuestreo pura: se construye un conjunto de entrenamiento, con la misma cardinalidad del original, extrayendo las muestras con reemplazo, mientras que los patrones no extraídos (aproximadamente el 36,8% de los datos, en media) se utiliza como conjunto de retención (Anguita, Ghio, Greco, Oneto, & Ridella, 2010).

Supuestos o Condiciones para el uso seguro de K – Fold Cross Validation en las pruebas paramétricas

La distinción entre pruebas paramétricas y no paramétricas se basa en el nivel de medida que representan los datos que se van a analizar. Es decir, una prueba paramétrica suele utilizar datos compuestos por valores reales.

Sin embargo, esto último no implica que cuando dispongamos siempre de este tipo de datos, debamos utilizar un test paramétrico. Deben cumplirse otros supuestos iniciales para un uso seguro de las pruebas paramétricas. El incumplimiento de estas condiciones puede hacer que un análisis estadístico pierda credibilidad.

Las siguientes condiciones son necesarias para realizar con seguridad las pruebas paramétricas:

- Independencia: En estadística, es cuando el hecho de que ocurra un evento no cambia la probabilidad de otro evento, dos eventos son independientes.
- Normalidad: cuando el comportamiento observado sigue una distribución normal o gaussiana con una media específica μ y varianza σ, la observación es normal. La prueba de normalidad aplicada a la muestra puede indicar si esta condición existe en los datos observados. Normalmente se utilizan tres pruebas de normalidad para comprobar si existe normalidad.:
    - Kolmogorov-Smirnov: compara la distribución acumulada de los datos observados con la distribución acumulada esperada de una distribución gaussiana, obteniendo el valor p en función de ambas discrepancias.

- Shapiro-Wilk: analiza los datos observados para calcular el nivel de simetría y curtosis (forma de la curva) para calcular después la diferencia con respecto a una distribución gaussiana, obteniendo el valor p a partir de la suma de los cuadrados de las discrepancias. cuadrados de las discrepancias.

- D'Agostino-Pearson: primero calcula la asimetría y la curtosis para cuantificar lo lejos que está la distribución de Gauss en términos de asimetría y forma. A continuación, calcula en qué medida cada uno de estos valores difiere del valor esperado con una distribución gaussiana, y calcula un único valor p a partir de la suma de las discrepancias.

- Heteroscedasticidad: Esta propiedad indica la existencia de una violación de la hipótesis de igualdad de varianzas. La prueba de Levene se utiliza para comprobar si k muestras presentan o no homogeneidad de varianzas (homocedasticidad). Cuando los datos observados no cumplen la condición de normalidad, el resultado de esta prueba es más fiable que la prueba de Bartlett, que comprueba la misma propiedad.

Con respecto a la condición de independencia, la independencia no se verifica realmente en k-FCV y 5×2CV (una parte de las muestras se utiliza para el entrenamiento y la prueba en diferentes particiones). Las particiones de retención pueden tomarse con seguridad como independientes, ya que las particiones de entrenamiento y de prueba no se solapan.

La independencia de los eventos en términos de obtención de resultados suele ser obvia, dado que son ejecuciones independientes del algoritmo con semillas iniciales generadas aleatoriamente (Salvador, Luengo y Herrera, 2015).

2.4 Repeated k-fold Cross Validation

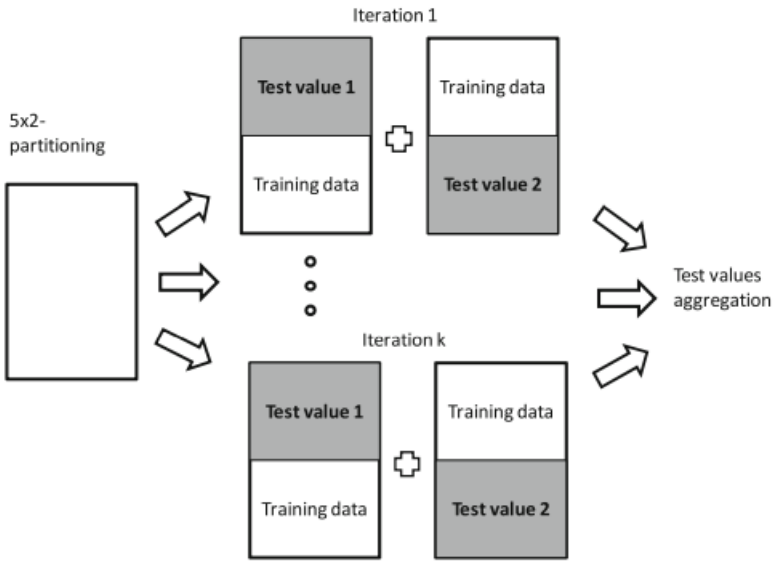
K-fold repetido es la técnica de validación cruzada más preferida para los modelos de aprendizaje automático de clasificación y regresión. La mezcla y el muestreo aleatorio del grupo de datos repetida varias veces es el procedimiento principal del algoritmo K-fold repetido y da como resultado la creación de un modelo robusto, ya que cubre las operaciones máximas de entrenamiento y prueba.

El funcionamiento de esta técnica de validación cruzada para evaluar la precisión de un modelo de aprendizaje automático depende de 2 parámetros. El primer parámetro es K, que es un valor entero y establece, que la información se dividirá en K pliegues (o subconjuntos). Entre los pliegues K, el modelo se entrena en los subconjuntos K-1 y el subconjunto restante se utilizará para evaluar el

rendimiento del modelo. Estos pasos se repetirán hasta un cierto número de veces que será decidido por el segundo parámetro de este algoritmo y, por lo tanto, recibió el nombre de K-fold Repetido, es decir, el algoritmo de validación cruzada de K-fold se repite un cierto número de veces.

Figura 8.

Proceso de 5x2 – Fold.



Pasos involucrados en la validación cruzada repetida de K-fold:

Cada iteración del K-fold repetido es la implementación de un algoritmo normal de K-fold. En la técnica de validación cruzada de K-fold están involucrados los siguientes pasos:

1. Divida los datos iniciales en K subconjuntos aleatorios.
2. Para cada uno de los subconjuntos desarrollados de puntos de datos
    - Trate ese subconjunto como el conjunto de validación
    - Utilice todos los subconjuntos restantes para fines de entrenamiento
    - Entrenamiento del modelo y evaluación en el conjunto de validación o conjunto de prueba
    - Calcular el error de predicción
3. Repita el paso anterior K veces, es decir, hasta que el modelo no esté entrenado y probado en todos los subconjuntos
4. Genere un error de predicción general tomando el promedio de los errores de predicción en cada caso.

Por lo tanto, en el método de validación cruzada repetida de k veces, los pasos anteriores se repetirán en el conjunto de datos dado durante un cierto número de veces. En cada iteración, habrá una división completamente diferente en K-folds y la puntuación de rendimiento del modelo también será diferente. Por último, la puntuación media de rendimiento en todos los casos dará la precisión final del modelo. Para llevar a cabo estas complejas tareas del método repetido de K-fold, el lenguaje R proporciona una rica biblioteca de funciones y paquetes incorporados. A continuación, se muestra el enfoque paso a paso para implementar la técnica de validación cruzada K-fold repetida en el modelo de aprendizaje automático de clasificación y regresión.

Implementar la validación cruzada repetida de K-fold en la clasificación

Cuando la variable de destino es de tipo de datos categóricos, se utilizan modelos de aprendizaje automático de clasificación para predecir las etiquetas de clase. En este ejemplo, el algoritmo Naive Bayes se utilizará como clasificador probabilístico para predecir la etiqueta de clase de la variable objetivo.

Paso 1: cargar los paquetes y las bibliotecas necesarios

Se deben importar todas las librerías y paquetes necesarios para realizar la tarea sin ningún error.

Paso 2: explorar los datos

Si hay un caso de desequilibrio de clases en la variable de destino, se utilizan los siguientes métodos para corregirlo:

- Muestreo descendente
- Muestreo ascendente
- Muestreo híbrido con SMOTE y ROSE

Paso 3: construir el modelo con el algoritmo repetido de K-fold

La función trainControl () está definida para establecer el número de repeticiones y el valor del parámetro K. Después de eso, el modelo se desarrolla según los pasos involucrados en el algoritmo repetido de K-fold.

Paso 4: evaluar la precisión del modelo

En este paso final, se generará la puntuación de rendimiento del modelo después de probarlo en todos los pliegues de validación posibles. A continuación, se muestra el código para imprimir la precisión y el resumen general del modelo desarrollado.

Implementar la validación cruzada repetida de K-fold en la regresión

Los modelos de aprendizaje automático de regresión se prefieren para aquellos conjuntos de datos en los que la variable objetivo es de naturaleza continua, como la temperatura de un área, el costo de un producto básico, etc. Los valores de la variable objetivo son números enteros o de punto flotante. A continuación, se muestran los pasos necesarios para implementar el algoritmo de k-veces repetido como técnica de validación cruzada en modelos de regresión.

Paso 1: cargar la información y los paquetes necesarios

Como primer paso, el entorno R debe cargarse con todos los paquetes y bibliotecas esenciales para realizar diversas operaciones.

Paso 2: cargar e inspeccionar la información

Una vez que se importan todos los paquetes, es hora de cargar el dataset deseado. Para construir un modelo correcto, es necesario conocer la estructura del conjunto de datos.

Paso 3: construir el modelo con el algoritmo repetido de K-fold

La función trainControl () está definida para establecer el número de repeticiones y el valor del parámetro K. Después de eso, el modelo se desarrolla según los pasos involucrados en el algoritmo repetido de K-fold.

Paso 4: evaluar la precisión del modelo

Según el algoritmo de la técnica K-fold repetida, ese modelo se prueba con cada pliegue (o subconjunto) único de la información original y, en cada caso, se calcula el error de predicción y, por último, la media de todos los errores de predicción se trata como el resultado final puntuación de rendimiento del modelo.

Ventajas de la validación cruzada repetida de K-fold

En un método muy eficaz para estimar el error de predicción y la precisión de un modelo.

En cada repetición, la muestra de datos se baraja, lo que da como resultado el desarrollo de diferentes divisiones de los datos de la muestra.

Desventajas de la validación cruzada repetida de K-fold

Un valor más bajo de K conduce a un modelo sesgado, y un valor más alto de K puede conducir a una variabilidad en las métricas de desempeño del modelo. Por lo tanto, es esencial utilizar el valor correcto de K para el modelo (generalmente es deseable K = 5 y K = 10).

Con cada repetición, el algoritmo tiene que entrenar el modelo desde cero, lo que significa que el tiempo de cálculo para evaluar el modelo aumenta con los tiempos de repetición.

Para reducir la varianza de la medida de rendimiento estimada, la validación cruzada se repite a veces con subconjuntos k-fold diferentes (rtimes repeated k-fold cross-validation). Sin embargo, Molinaro et al. demostraron que tales repeticiones reducen la varianza sólo ligeramente (Berrar, 2018).

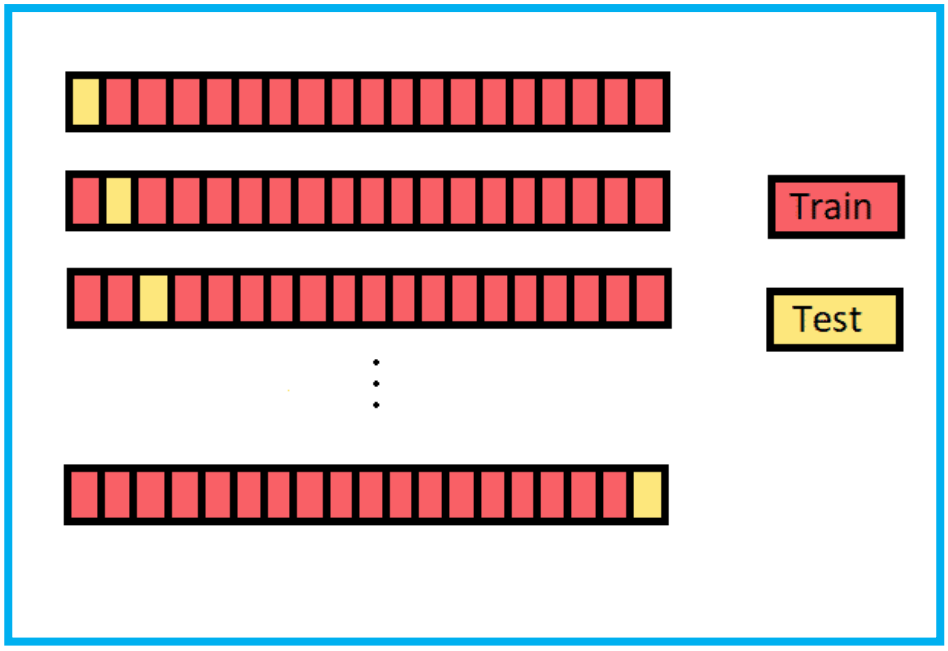
2.5 Leave One Out Cross-Validation (LOOCV)

El método LOOCV es un método iterativo: consiste en considerar todas las observaciones como conjunto de entrenamiento excepto una, la cual se utilizará como validación. Este proceso se repite hasta que cada observación se haya tomado como validación y las demás observaciones como el conjunto de entrenamiento. Es así que la variabilidad del error depende de la observación tomada como validación.

Para el cálculo del error del LOOCV, es el promedio de todos los errores calculados en cada interacción.

Figura 9.

Proceso de partición del dataset en entrenamiento (training) y validación (testing) según LOOCV.



"LOOCV es una versión extrema de la validación cruzada de k-fold que tiene el costo computacional máximo".

Se recomienda su uso

Conjuntos de datos pequeños o cuando el rendimiento estimado del modelo sea crítico.

No se recomienda su uso

Para dataset que presentan gran volumen o se aplican modelos costosos de adaptar. Dada la estimación mejorada del rendimiento del modelo.

Este caso, en particular, cuando la información tiene pocas observaciones, como menos de miles de ejemplos, puede provocar un sobreajuste del modelo durante el entrenamiento y estimaciones sesgadas del rendimiento del modelo.

Además, dado que no se utiliza muestreo de la data de entrenamiento, este procedimiento de estimación es determinista, a diferencia de las divisiones de prueba de tren y otras confirmaciones de validación cruzada de k veces que proporcionan una estimación estocástica del rendimiento del modelo.

Consideraciones de los modelos de re-muestreo

No existe un método de validación que supere al resto en todos los escenarios, la elección debe basarse en varios factores.

Si el tamaño de la muestra es pequeño, se recomienda emplear repeated k-Fold-Cross-Validation, ya que consigue un buen equilibrio sesgo-varianza y, dado que no son muchas observaciones, el coste computacional no es excesivo.

Si el objetivo principal es comparar modelos más que obtener una estimación precisa de las métricas, se recomienda bootstrapping ya que tiene menos varianza.

Si el tamaño muestral es muy grande, la diferencia entre métodos se reduce y toma más importancia la eficiencia computacional. En estos casos, 10-Fold-Cross-Validation simple es suficiente. (Kuhn, M & Johnson, K.,2013).

Tabla 01.

Ventajas y desventajas de las técnicas de re-muestreo.

MÉTODOS	Ventajas	Desventajas
Bootstrap	No requiere de hipótesis sobre el mecanismo generador de datos. Entonces, no requiere de fórmula matemática teórica para la selección de muestra.	Requiere de recursos computacionales robustos. Esto era un inconveniente más común en los años 80' por la escasez de los recursos computacionales. Pero, en nuestra época, sigue siendo un inconveniente en bases de datos pesadas o con un gran número de muestras.
	El bootstrap no paramétrico no requiere de suposición de distribuciones teóricas.	Necesita desarrollar programas de ordenador adecuadas a las circunstancias particulares de cada caso.
	En la mayoría de casos, permite aproximar el sesgo y varianza de un análisis estadístico con mayor precisión	Puede fallar debido a la presencia de casos atípicos
	El método de remuestreo bootstrap proporciona estimadores más robustos	Así como otros métodos existentes, es difícil pueda resolver problemas derivados de tamaños de muestra muy pequeños o malos diseños de muestreo.
K - Fold Cross Validation	Cada subconjunto se utiliza sucesivamente como conjunto de validación, mientras que los otros subconjuntos se utilizan para entrenar la superficie de respuesta.	El método descrito es un procedimiento de división parcial de los datos, ya que no se utilizan todas las combinaciones posibles de subconjuntos de validación.
	La validación cruzada de K pliegues es un estimador mucho mejor del rendimiento de nuestro modelo, incluso más que nuestra división entrenamiento-prueba.	Los diagramas muestran claramente que la validación cruzada de k pliegues tiene un sesgo pesimista, especialmente para dos y cinco pliegues.
	La mayoría de las estimaciones son razonablemente buenas a 10 pliegues y a 20 pliegues son casi insesgadas.	
Repeated K - Fold Cross Validation	En un método muy eficaz para estimar el error de predicción y la precisión de un modelo.	Un valor más bajo de K conduce a un modelo sesgado, y un valor más alto de K puede conducir a una variabilidad en las métricas de desempeño del modelo. Por lo tanto, es esencial utilizar el valor correcto de K para el modelo (generalmente es deseable K = 5 y K = 10).
	En cada repetición, la muestra de datos se baraja, lo que da como resultado el desarrollo de diferentes divisiones de los datos de la muestra.	Con cada repetición, el algoritmo tiene que entrenar el modelo desde cero, lo que significa que el tiempo de cálculo para evaluar el modelo aumenta con los tiempos de repetición.
	La repetición de la validación cruzada varias veces utiliza diferentes divisiones en pliegues por lo que proporciona una mejor estimación de Monte-Carlo a la validación cruzada completa con un coste añadido.	
Leave One Out Cross Validation	Por pruebas realizadas funciona adecuadamente para funciones de error continuas, como el error cuadrático medio.	En caso de conjunto de datos pequeño, puede provocar sobreajuste del modelo durante el entrenamiento y estimaciones sesgadas del rendimiento del modelo.
	El método no se ve afectado por el método de dividir el conjunto de prueba y el conjunto de verificación, porque cada información se ha probado por separado.	"LOOCV es una versión extrema de la validación cruzada de k-fold que tiene el costo computacional máximo.

III. METODOLOGÍA

3.1 Tipo de investigación

Enfoque: Cuantitativo. Emplea uso de información cuantitativa o cualitativa que puede ser cuantificable, por lo tanto, emplearse estadística para su análisis e interpretación de resultados (Hernández & Mendoza, 2018).

Tipo: Aplicada. Se emplearán teorías creadas por diversos autores, llevando a cabo su aplicación al área de estudio.

Nivel: Descriptivo. Indaga en las incidencias de las modalidades, categorías o niveles de una o más variables de una población. Buscan desarrollar una imagen o fiel representación (descripción) del fenómeno estudiado a partir de sus características. Describir en este caso es sinónimo de medir.

3.2 Diseño de investigación

No experimental.

Análisis secundario de datos. Se empleó un conjunto de datos obtenido de un experimento inicial y se ha empleado esta información en nuevos objetivos.

3.3 Técnicas e instrumentos de investigación

El trabajo se basa en una investigación documental, que consiste en presentar métodos de remuestreo como: Bootstrap, K-fold validación cruzada, K-fold validación cruzada repetido, Leave One Out Cross-Validation. Bajo un diseño de investigación cuantitativo y descriptivo. El cual tiene como objetivo mostrar que método tiene una mayor performance para validar nuestro modelo de regresión múltiple. Las métricas halladas en cada método es la estimación de los coeficientes de regresión ($\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{\beta}_2$) y el error estándar de cada coeficiente.

3.4 Procedimientos

Para el presente trabajo nos basaremos en una data real de la que se extrajo una muestra. La cual está constituida por 100 observaciones y para solo se trabajará con tres variables:

- $Y =$ kg.ha : $kg.ha^{-1}$ de bayas de arándano cosechadas por semana en una válvula del fundo.
- $X_1 =$ BayasCosechables: promedio del número de frutos cosechables por planta registrados por semana en una válvula.

- $X_2 =$ Hcalor : Acumulado de horas de calor (con temperatura mayor a los 24 °C) registrados en la semana.

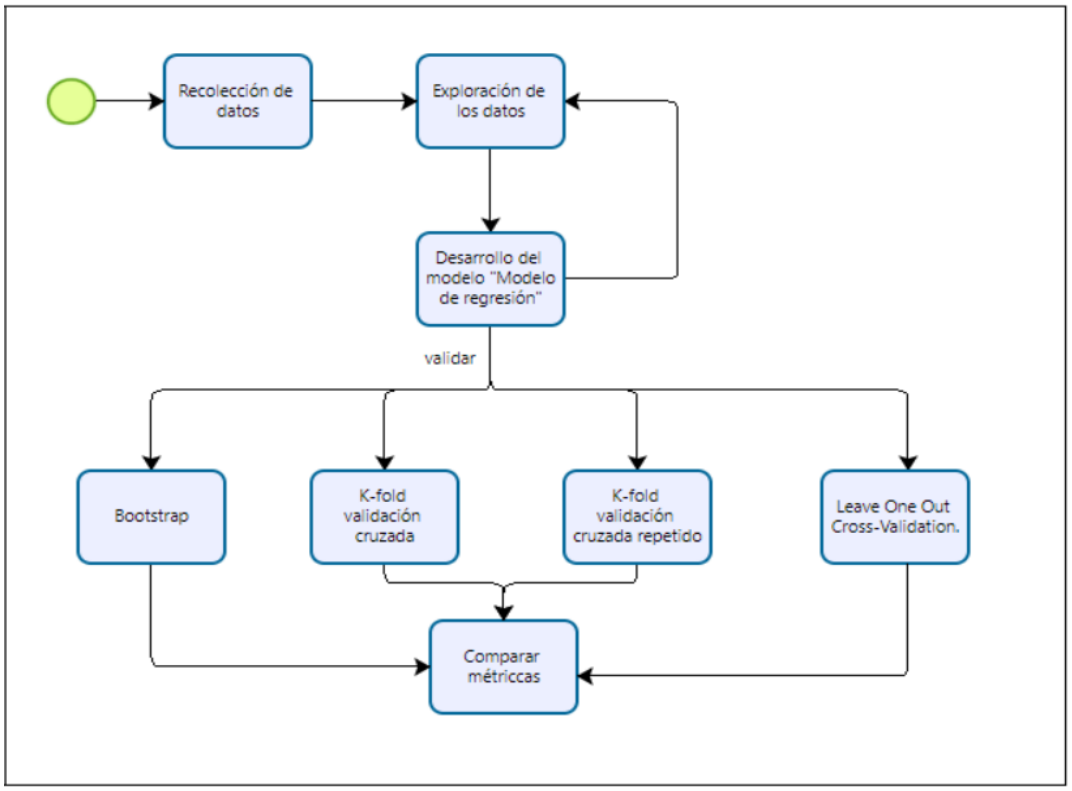
Los datos se obtuvieron de la empresa Agrícola Cerro Prieto S.A en la campaña agrícola del cultivo de arándano 2020-2021. Fundo ubicado en Chepén (La Libertad) que cuenta con 490 hectáreas aproximadamente.

La muestra fue seleccionada mediante un muestreo aleatorio simple. Antes del planteo del modelo de regresión lineal se realizó un análisis exploratorio de los datos encontrándose una relación directa entre las variables Bayas cosechables y Kg cosechados y una relación inversa entre variables Horas de calor y Kg cosechados.

El procedimiento y análisis de los datos se realizó empleando el software estadístico R 4.0.1 y la interfaz gráfica RStudio.

Figura 10

Diagrama de flujo de los procedimientos empleados.



IV. RESULTADOS

Según las medidas descriptivas, se observa que la media de la variable Bayas Cosechables es de 80.58 frutos por planta en una semana, y que la media del acumulado de horas de calor (HCalor) es 21.82 unidades, y por último se tiene que la media de la variable rendimiento de frutos de arándanos cosechadas por hectárea es de 385.63 $kg.ha^{-1}$ en una válvula del fundo.

Tabla 02.

Descriptivos de las variables de los datos de estudio.

	BayasCosechables	Hcalor	Kg.ha
Mean	80.58	21.82	385.63
Std.Dev	48.22	21.60	255.97
Min	0.00	0.00	28.39
Q1	49.81	2.50	151.20
Median	77.94	13.50	351.22
Q3	106.21	38.00	558.94
Max	212.67	75.00	1126.82
MAD	41.82	17.05	303.79
IQR	56.14	35.25	397.59
CV	0.60	0.99	0.66
Skewness	0.56	0.78	0.60
SE.Skewness	0.24	0.24	0.24
Kurtosis	0.21	-0.53	-0.32
N.Valid	100.00	100.00	100.00
Pct.Valid	100.00	100.00	100.00

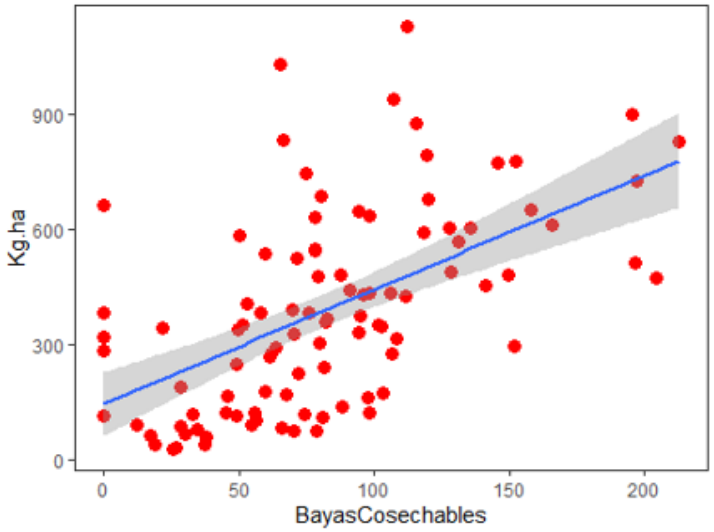
En el siguiente gráfico de diagrama de dispersión (figura 11), se aprecia que las observaciones del número de bayas cosechables vs los Kg Cosechados por hectárea son muy dispersos asi como también se observa que tenemos valores atípicos, en el número de Bayas Cosechables aproximadamente igual a 60 se tienen más de 1000 Kg cosechados por hectárea. Pero estos casos con menor número de bayas y una mayor productividad se pueden deber a que el peso promedio de los frutos en esas válvulas (observaciones) es mayor, por un comportamiento agronómico propio de la variedad. Las plantas de determinadas variedades de arándano y la mayoría de cultivos que presentan menor número de frutos, presentan regularmente un mayor peso promedio de frutos.

Las observaciones más altas los puntos se alejan más de la recta, pero podríamos suponer que los datos siguen una distribución lineal. Por otro lado, se puede observar que la dispersión de la

respuesta “Y” observada, tiene mayor error con respecto a la media condicional de la recta de regresión, a medida que el valor de x aumenta, lo que sería evidencia de un incumplimiento del supuesto de homocedasticidad.

Figura 10.

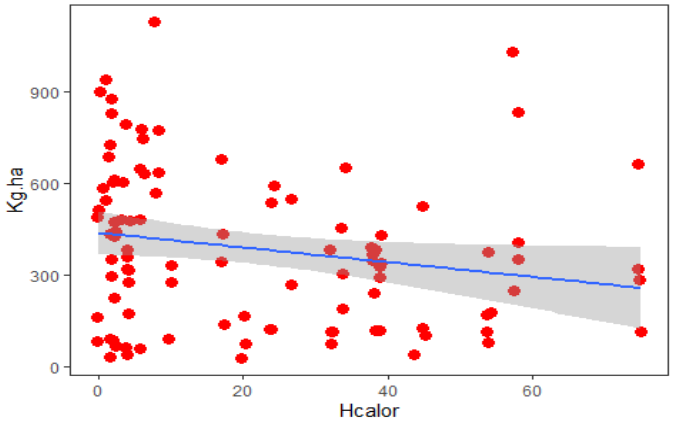
Diagrama de dispersión de variables Bayas cosechables y Kg cosechados.



Asimismo, se puede observar en la figura 12, que el acumulado de horas de calor (mayores de 24 °C) por semana tienen una relación inversa con los Kg.ha de bayas de arándano registrados por válvula, lo que supone que, a medida que el clima se torne más frío (temporada de otoño - invierto) la productividad del arándano es mayor.

Figura 11.

Diagrama de dispersión de variables Horas de calor y Kg cosechados.



Bajo los datos analizados, se planteó el siguiente modelo de regresión lineal múltiple:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2$$

Donde:

$X_1 : Bayas\ cosechables$

$X_2 : Horas\ de\ calor$

$\hat{\beta}_0 : Intercepto$

$\hat{\beta}_1 : Coeficiente\ de\ Bayas\ cosechables$

$\hat{\beta}_2 : Coeficiente\ de\ Horas\ de\ calor$

Aplicando las técnicas y métodos indicados inicialmente, tenemos los siguientes resultados:

Tabla 03.

Comparación de resultados de técnicas de re-muestreo y modelo clásico de regresión lineal.

Método	$\widehat{\boldsymbol{\beta}}_{\boldsymbol{0}}$	$\widehat{\boldsymbol{\beta}}_{\boldsymbol{1}}$	$\widehat{\boldsymbol{\beta}}_{\boldsymbol{2}}$	$\boldsymbol{R^2}$ ***ajustado***
Regresión lineal	112.507 (60.08)	3.154(0.50)	0.866 (1.87)	0.304
Bootstrap	118.954 (48.09)	3.153 (0.37)	0.485 (1.05)	0.303 (0.07)
K - Fold CV	101.106	3.251	0.611	0.349
K - Fold CV Repetido	106.626	3.268	0.479	0.342
LOOCV	112.504 (53.23)	3.154 (0.42)	0.865 (1.26)	0.304 (0.07)

Nota. Los valores del error estándar de cada coeficiente y estadístico están entre paréntesis según corresponda.

La interpretación de los coeficientes estimados del modelo de regresión lineal es:

- $\hat{\beta}_0$: En nuestro caso, el intercepto no tiene interpretación coherente con la aplicación dado que si el número de bayas cosechas promedio por planta es cero entonces una válvula no tendría rendimiento.

- $\hat{\beta}_1$: Por cada incremento de una unidad de BayasCosechables, el promedio de Kg.ha de bayas de arándano cosechadas por semana en una válvula del fundo aumenta en 3.1548

(unidades), luego de mantener fijo las demás variables. Con un nivel de significancia de 5%, esta relación es altamente significativa (p=9.53e-09).

- $\hat{\beta}_2$: Por cada incremento de una unidad de Hcalor, el promedio de Kg.ha de bayas de arándano cosechadas por semana en una válvula del fundo aumenta en 0.8662 (unidades), luego de mantener fijo las demás variables. Con un nivel de significancia de 5%, esta relación no es significativa (p=0.4416).

Por otro lado, del resultado de los modelos aplicando re-muestreo, tenemos lo siguiente:

- El método de re-muestreo más optimista fue es K - Fold CV con un R^2 ajustado de 0.349.
- El método de re-muestreo pesimista fueron el LOOCV y Bootstrap con un R^2 ajustado de 0.304 y 0.303 respectivamente.
- El método de re-muestro válido fue el K - Fold CV Repetido con un R^2 ajustado de 0.342.

V. CONCLUSIONES

1. En el presente trabajo, consideramos que el método de re-muestreo válido para la aplicación del rendimiento de frutos por hectárea según la influencia del número de bayas cosechables por planta y de las horas de calor acumuladas por semana, fue el método de K - Fold CV Repetido por tener un R^2 ajustado óptimo de 0.342.

2. En el presente trabajo, consideramos que el método de re-muestreo más optimista para la aplicación del rendimiento de frutos por hectárea según la influencia del número de bayas cosechables por planta y de las horas de calor acumuladas por semana fue el método de K - Fold CV con un R^2 ajustado de 0.349.

3. En el presente trabajo, consideramos que el método de re-muestreo pesimista para la aplicación del rendimiento de frutos por hectárea según la influencia del número de bayas cosechables por planta y de las horas de calor acumuladas por semana fueron el método de LOOCV y el Bootstrap con un R^2 ajustado de 0.304 y 0.303 respectivamente.

VI. RECOMENDACIONES

1. La forma correcta de aplicar el método de remuestreo Cross-validation (CV), este puede ser usado para dos propósitos diferentes: 1) seleccionar variables, y 2) estimar el error de predicción este debe realizarse para ambos objetivos, tanto para la selección de variables en una primera fase, como para la estimación del error de predicción, en la segunda fase.

2. Los diferentes métodos de remuestreo nos confirman que la variable **Hcalor** (horas de calor) no tiene un aporte significativo para estimar la variable respuesta. Por tanto, para modelos posteriores recomendamos no utilizar esta variable.

3. Con el propósito de obtener el mejor modelo (mayor bondad de ajuste) para nuestros datos, recomendaríamos realizar una aplicación de prueba de predicción a otra muestra de validación, para así identificar si algunos de los métodos de re-muestreo generan estimaciones o modelos sobreajustados.

VII. REFERENCIAS

Anguita, D., Ghio, A., Greco, N., Oneto, L., & Ridella, S. (2010, July). Model selection for support vector machines: Advantages and disadvantages of the machine learning theory. In *The 2010 international joint conference on neural networks (IJCNN)* (pp. 1-8). IEEE.

Anonimo (s.f.) Repeated K-fold Cross Validation in R Programming. Recuperado de: https://www.geeksforgeeks.org/repeated-k-fold-cross-validation-in-r-programming/

Anonimo (s.f.). Resampling Methods. Recuperado de: https://uc-r.github.io/resampling_methods

Berrar, D. (2018). Cross-Validation. *Encyclopedia of Bioinformatics and Computational Biology*, Volume 1, Elsevier, pp. 542–545. https://doi.org/10.1016/B978- 0-12- 809633-8.20349- X

Beschorner, A., Voigt, M., & Vogeler, K. (2014). Monte carlo cross-validation for response surface benchmark. *In 12th International Probabilistic Workshop.*

Efron, B. (1979). *"Bootstrap Methods: Another Look at the Jackknife".* Institute of Mathematical Statistics.

Efron, B. (1982). The jackknife, the bootstrap, and other resampling plans, In *Society of Industrial and Applied Mathematics.*

Efron, B. y Tibshirani, R. (1993). "*An introduction to the bootstrap*". Springer Science and Business Media Dordrecht.

García, S., Luengo, J. y Herrera, F. (2015) "*Data preprocessing in Data Mining".* Springer.

Gujarati, D. y Porter, D. (2010). "*Econometría*". McGraw Hill.

Hernández, R., & Mendoza, C. P. (2018). *Metodología de la Investigación. Las rutas cuantitativa, cualitativa y mixta.* Ciudad de México.: Mc GRAW-HILL INTERAMERICANA EDITORES, S.A. de C.V.

Kohavi, R. (2001). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. Recuperado de: https://www.researchgate.net/publication/2352264_A_Study_of_Cross-Validation_and_Bootstrap_for_Accuracy_Estimation_and_Model_Selection

Kuhn, M & Johnson, K. (2013). Applied Predictive Modeling. *Splinger.*

Levin, R. y Rubin, D. (2004). "*Estadística para administración y economía*". Pearson.

McKinney, W. (2018). "*Python for Data Analysis*". O' Reilly Media.

Ozdemir, S. (2016). Principles of Data Science: Learn the techniques and math you need to start making sense of your data. *Packt Publishing*.

Quenouille, M. (1949). Approximate test of correlation in time series. J.R. Statist.

Salvador García, Julián Luengo y Francisco Herrera (2015). Data Preprocessing in Data Mining. *Springler*.

VIII. ANEXOS

Anexo 1: Código en R de lectura de datos

```
datos <- readxl::read_xlsx("Cosecha arandano.xlsx", sheet = "Hoja1")
str(datos)
```

```
## tibble [6,745 x 10] (S3: tbl_df/tbl/data.frame)
##  $ Variedad             : chr [1:6745] "Biloxi" "Biloxi" "Biloxi" "
Biloxi" ...
##  $ SemCal               : num [1:6745] 45 46 47 48 49 25 25 25 26 2
5 ...
##  $ KgCosechados         : num [1:6745] 71.8 114.8 145 66.9 96.2 ...
##  $ BayasiniciandoaCremas: num [1:6745] 0 0 0 0 29.6 ...
##  $ BayasCremas          : num [1:6745] 0 0 0 0 0 ...
##  $ BayasMaduras         : num [1:6745] 0 0 0 0 0 ...
##  $ BayasCosechables     : num [1:6745] 0 0 0 0 29.6 ...
##  $ Hectareas            : num [1:6745] 2.72 2.72 2.72 2.72 2.72 2 3
.2 3.2 2.46 3.2 ...
##  $ Plantasprod          : num [1:6745] 8086 8086 8086 8086 8086 ...
##  $ Hcalor               : num [1:6745] 38 39 27 45 54 20 20 20 44 2
0 ...
```

```
library(dplyr)
```

```
datos <- mutate(datos, Kg.ha = KgCosechados/Hectareas)
```

```
datos <- datos %>%
  filter(!Kg.ha %in% "0")
```

```
datos <- datos[,c(7,10,11)]
```

```
datos <- na.omit(datos)
RNGkind(sample.kind = "Rounding")
```

```
set.seed(100)
datos <- datos[sample(nrow(datos),100,replace = F),]
datos
```

```
## # A tibble: 100 x 3
##    BayasCosechables Hcalor Kg.ha
##               <dbl>  <dbl> <dbl>
##  1             77.9     27 550.
##  2             30.2      2  69.2
##  3             75.8      4 384.
##  4             25.5     20  28.4
##  5             94.5     10 332.
##  6             51.2     58 353.
##  7            118.      24 591.
##  8             91.0      2 444.
##  9            131.       8 568.
## 10             37.6      6  60.6
## # ... with 90 more rows
```

Anexo 2: Código en R del análisis exploratorio y descriptivo

```
library(summarytools)

summarytools::descr(datos)
```

```
## Descriptive Statistics
## datos
## N: 100
##
##                      BayasCosechables   Hcalor     Kg.ha
## ----------------- ------------------ -------- ---------
##              Mean              80.58    21.82    385.63
##           Std.Dev              48.22    21.60    255.97
##               Min               0.00     0.00     28.39
##                Q1              49.81     2.50    151.20
##            Median              77.94    13.50    351.22
##                Q3             106.21    38.00    558.94
##               Max             212.67    75.00   1126.82
##               MAD              41.82    17.05    303.79
##               IQR              56.14    35.25    397.59
```

```
##                CV                0.60      0.99       0.66
##          Skewness                0.56      0.78       0.60
##       SE.Skewness                0.24      0.24       0.24
##          Kurtosis                0.21     -0.53      -0.32
##           N.Valid              100.00    100.00     100.00
##         Pct.Valid              100.00    100.00     100.00
```

```
#Generando gráfico de dispersión del número de bayas cosechables vs lo
s Kg Cosechados por hectárea
library(ggplot2)
```

```
datos %>%
  ggplot(aes(x=BayasCosechables,y=Kg.ha))+
  geom_point(position = "jitter", size=3, colour="red")+
  labs( title = "DIAGRAMA DE DISPERSIÓN")+
  geom_smooth(method = "lm")+
  theme_test()
```

```
#Generando gráfico de dispersión del número de bayas cosechables vs lo
s Kg Cosechados por hectárea
datos %>%
  ggplot(aes(x=Hcalor,y=Kg.ha))+
  geom_point(position = "jitter", size=3, colour="red")+
  labs( title = "DIAGRAMA DE DISPERSIÓN")+
  geom_smooth(method = "lm")+
  theme_test()
```

Anexo 3: Código en R de aplicación modelo clásico de regresión lineal

```
summary(lm(Kg.ha~BayasCosechables+Hcalor,datos))
```

```
##
## Call:
## lm(formula = Kg.ha ~ BayasCosechables + Hcalor, data = datos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -318.59 -167.46  -39.73  118.38  661.05
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      112.5071    60.0894   1.872   0.0642 .
## BayasCosechables   3.1548     0.5023   6.281 9.53e-09 ***
## Hcalor             0.8662     1.1212   0.773   0.4416
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 213.5 on 97 degrees of freedom
## Multiple R-squared:  0.3182, Adjusted R-squared:  0.3042
## F-statistic: 22.64 on 2 and 97 DF,  p-value: 8.544e-09
```

Anexo 4: Código en R de aplicación de bootstrap

```
boot3<-function(datos,B,estadistico,...){
  n<-nrow(datos)
  p<-ncol(datos)
  estaboot<-matrix(0,B,p)
  for(i in 1:B){
    indices<-sample(1:n,n,T)
    estaboot[i,]<-estadistico(datos[indices,],...)
  }

  esboot<-apply(estaboot,2,mean)
  eeboot<-apply(estaboot,2,sd)
  return(list(esboot=esboot,eeboot=eeboot))
}

coefi<-function(datos,y){
  datos<-as.matrix(datos)
  betas<-lm(datos[,y]~datos[,-y])$coe
  return(betas)
}
```

```r
boot1<-function(datos,B,estadistico,...){
  n<-nrow(datos)
  p<-ncol(datos)
  estaboot<-matrix(0,B,p)
  for(i in 1:B){
    indices<-sample(1:n,n,T)
    estaboot[i,]<-estadistico(datos[indices,],...)
  }

  esboot<-mean(estaboot)
  eeboot<-sd(estaboot)
  return(list(esboot=esboot,eeboot=eeboot))
}


r2adj<-function(datos,y){
  datos<-as.matrix(datos)
  r2adj <- summary(lm(datos[,y]~datos[,-y]))$adj.r.squared
  return(r2.adj = r2adj)
}

RNGkind(sample.kind="Rounding")

set.seed(99)
boot3(datos,50,coefi,3)

## $esboot
## [1] 118.9549832   3.1534385   0.4859167
##
## $eeboot
## [1] 48.0906795  0.3733184  1.0592750

boot1(datos,50,r2adj,3)
```

```
## $esboot
## [1] 0.3033086
##
## $eeboot
## [1] 0.0711541
```

Anexo 5: Código en R de aplicación de validación cruzada

```
crossval<-function(datos,K,r,d){
  datos<-as.matrix(datos)
  n<-nrow(datos)
  EVC<-c()
  resid<-c()
  subm<-floor(n/K)
  resi<-lm(datos[,d]~datos[,-d])$res
  APE<-sum(resi^2)/n
  for(i in 1:r){
    indices<-sample(n,n)
    azar<-datos[indices,]

    for(j in 1:K){
      unid<- ((j-1)*subm+1):(subm*j)
      if (j==K)
      {
        unid<-((j-1)*subm+1):n
      }
      datosp<-azar[unid,]
      datose<-azar[-unid,]
      ye<-datose[,d]
      xe<-datose[,-d]
      betas<-lm(ye~xe)$coef
      r2adj <- summary(lm(ye~xe))$adj.r.squared
      datosp1<-cbind(1,datosp[,-d])
      estim<-datosp1%*%betas
      resid[j]<-sum((datosp[,d]-estim)^2)
```

```
    }
    EVC[i]<-sum(resid)/n
  }
  EVCP<-mean(EVC)
  cvEVC<-sd(EVC)*100/EVCP
  sesgo<-EVCP-APE
  return(list(betas = betas, r2.adj = r2adj, APE=APE,EVCP=EVCP,cvEVC=c
vEVC,sesgo=sesgo))
}

RNGkind(sample.kind="Rounding")

set.seed(80)
crossval(datos,10,1,3)

## $betas
##        (Intercept) xeBayasCosechables            xeHcalor
##        101.1063511           3.2515176           0.6110509
##
## $r2.adj
## [1] 0.3495893
##
## $APE
## [1] 44223.92
##
## $EVCP
## [1] 46707.31
##
## $cvEVC
## [1] NA
##
## $sesgo
## [1] 2483.389
```

Anexo 6: Código en R de aplicación de validación cruzada repetida

```
crossval<-function(datos,K,r,d){
  datos<-as.matrix(datos)
  n<-nrow(datos)
  EVC<-c()
  resid<-c()
  subm<-floor(n/K)
  resi<-lm(datos[,d]~datos[,-d])$res
  APE<-sum(resi^2)/n
  for(i in 1:r){
    indices<-sample(n,n)
    azar<-datos[indices,]

    for(j in 1:K){
      unid<- ((j-1)*subm+1):(subm*j)
      if (j==K)
      {
        unid<-((j-1)*subm+1):n
      }
      datosp<-azar[unid,]
      datose<-azar[-unid,]
      ye<-datose[,d]
      xe<-datose[,-d]
      betas<-lm(ye~xe)$coef
      r2adj <- summary(lm(ye~xe))$adj.r.squared
      datosp1<-cbind(1,datosp[,-d])
      estim<-datosp1%*%betas
      resid[j]<-sum((datosp[,d]-estim)^2)
    }
    EVC[i]<-sum(resid)/n
  }
  EVCP<-mean(EVC)
  cvEVC<-sd(EVC)*100/EVCP
  sesgo<-EVCP-APE
  return(list(betas = betas, r2.adj = r2adj, APE=APE,EVCP=EVCP,cvEVC=c
vEVC,sesgo=sesgo))
```

```
}

RNGkind(sample.kind="Rounding")

set.seed(80)
crossval(datos,10,5,3)

## $betas
##         (Intercept) xeBayasCosechables            xeHcalor
##         106.6268053           3.2689411           0.4796198
##
## $r2.adj
## [1] 0.3428871
##
## $APE
## [1] 44223.92
##
## $EVCP
## [1] 46938.86
##
## $cvEVC
## [1] 1.39587
##
## $sesgo
## [1] 2714.946
```

Anexo 7: Código en R de aplicación de LOOCV

```
jack2<-function(datos,estadistico,...){
  n<-nrow(datos)
  estjack<-c()
  for(i in 1:n){
    estjack[i]<-estadistico(datos[-i,],...)
  }
  esjack<-mean(estjack)
  eejack<-(n-1)*sd(estjack)/sqrt(n)
```

```
  return(list(esjack=esjack,eejack=eejack))
}

coefi1<-function(datos,y){
  datos<-as.matrix(datos)
  betas<-lm(datos[,y]~datos[,-y])$coe
  return(Intercepto = betas[1])
}

coefi2<-function(datos,y){
  datos<-as.matrix(datos)
  betas<-lm(datos[,y]~datos[,-y])$coe
  return(beta1 = betas[2])
}

coefi3<-function(datos,y){
  datos<-as.matrix(datos)
  betas<-lm(datos[,y]~datos[,-y])$coe
  return(beta2 = betas[3])
}

r2adj<-function(datos,y){
  datos<-as.matrix(datos)
  r2adj <- summary(lm(datos[,y]~datos[,-y]))$adj.r.squared
  return(r2.adj = r2adj)
}


RNGkind(sample.kind="Rounding")

set.seed(99)
jack2(datos,coefi1,3)

## $esjack
## [1] 112.5046
```

```
##
## $eejack
## [1] 53.23454
```

```
jack2(datos,coefi2,3)
```

```
## $esjack
## [1] 3.154972
##
## $eejack
## [1] 0.426692
```

```
jack2(datos,coefi3,3)
```

```
## $esjack
## [1] 0.8656901
##
## $eejack
## [1] 1.264109
```

```
jack2(datos,r2adj,3)
```

```
## $esjack
## [1] 0.3041685
##
## $eejack
## [1] 0.07615932
```