TRABAJO FIN DE MÁSTER EN ESTADÍSTICA APLICADA

UNIVERSIDAD DE GRANADA

COMPARACIÓN DE REGRESIÓN LOGÍSTICA Y ÁRBOLES DE DECISIÓN EN DATOS BIBLIOMÉTRICOS

Autor

Carlos Alberto Ramos Soler

Tutor

Manuel Escabias Machuca

Departamento de Estadística e Investigación Operativa

Febrero de 2023

Tabla de Contenido

Lista de figuras

1. Resumen

La necesidad de cuantificar relaciones entre variables para explicar el comportamiento de una de ellas en función de otras ha generado el surgimiento de múltiples técnicas. En este sentido, los modelos de regresión logística surgen como una alternativa para modelar respuestas de tipo categórico, estos modelos forman parte de la familia exponencial de densidades.

La irrupción del Data Mining ha permitido que afloren diferentes métodos computacionales que sirven como alternativas para resolver el tratamiento y clasificación de este tipo de datos. Entre estas técnicas se encuentran los árboles de decisión, los cuales son métodos no paramétricos que no requieren supuestos distribucionales.

En este trabajo se realiza una comparación entre la regresión logística multinomial y los árboles de decisión. El conjunto de datos utilizado está relacionado con variables bibliométricas, correspondientes a registros de caracterización de autores de artículos científicos en el sector salud.

Abstract

The need to quantify relationships between variables to explain the behavior of one of them based on others has generated the emergence of multiple techniques. In this sense, logistic regression models emerge as an alternative to model categorical responses, these models are part of the exponential family of densities.

The irruption of Data Mining has allowed the emergence of different computational methods that serve as alternatives to solve the treatment and classification of this type of data. Among these techniques are decision trees, which are non-parametric methods that do not require distributional assumptions.

In this paper, a comparison is made between multinomial logistic regression and decision trees. The data set used is related to bibliometric variables, corresponding to characterization records of authors of scientific articles in the health sector.

2. Introducción

La actividad estadística a lo largo del tiempo ha buscado en diferentes campos del conocimiento cuantificar relaciones entre variables para explicar el comportamiento de una de ellas en función de otras. Este tipo de análisis normalmente se hace a través de un modelo de regresión o de diseño.

Según como lo expresa Díaz (2009), "Un modelo está compuesto por la variable a ser explicada (dependiente o respuesta) y las variables explicativas (independientes o regresoras), con las cuales se pretende dar cuenta del comportamiento de la variable respuesta". Los modelos se visualizan matemáticamente a través de una ecuación en la cual se expresa la relación entre la variable independiente y sus variables dependientes.

Ahora bien, cuando la variable respuesta es de tipo categórico, los modelos clásicos no son adecuados y se hace necesario recurrir a modelos más generales. Nelder y Wedderburn (1972) proponen los modelos lineales generalizados (GLM) como una extensión de los modelos lineales clásicos, e incorporan la modelación de variables categóricas en esta familia por medio de la regresión logística. En este trabajo se realiza una aplicación de estos modelos con respuesta multinomial.

Los modelos logísticos, como todos los que forman parte de la familia exponencial de densidades, son modelos paramétricos y deben cumplir con supuestos que en algunas ocasiones no son fáciles de cumplir. Es por esto por lo que han venido surgiendo alternativas para el modelamiento de este tipo de datos.

La irrupción del Data Mining ha hecho que afloren diferentes métodos computacionales para resolver el tratamiento y clasificación de datos categóricos. Entre estas técnicas que han surgido se encuentran los árboles de decisión, siendo estos métodos no paramétricos, que permiten detectar interacciones, modelar relaciones no lineales y no son sensibles a la presencia de datos faltantes y outliers ((Breiman, Friedman, Olshen & Stone, 1984), (Kass, 1980)). Su principio básico es generar particiones recursivas por reglas de

clasificación hasta llegar a una clasificación final, tal que es posible identificar perfiles (nodos terminales).

En este trabajo se realiza una comparación entre la regresión logística multinomial y los árboles de decisión en un conjunto de datos relacionados con variables bibliométricas que corresponden a registros de publicación de artículos científicos, el documento está organizado de la siguiente manera:

Inicialmente, en el capítulo 3 se presenta parte de la teoría relacionada con los modelos de regresión logística; en el apartado 3.1 se presenta el modelo de regresión logística simple, interpretación de parámetros y supuestos del modelo. En el apartado 3.2 se muestran generalidades de la regresión logística multinomial; formulación e interpretación, estimación y librería en R para el ajuste del modelo. El apartado 3.3 presenta teoría relacionada con árboles de decisión, ventajas y desventajas de estos y librerías para la construcción de los árboles en R. Finalmente, en el apartado 3.4 se presentan la tasa de clasificación errónea.

Posteriormente, en el capítulo 4 se presentan los resultados de la investigación. Este está organizado de la siguiente forma; El apartado 4.1 muestra un análisis descriptivo de las variables involucradas en la investigación. En el apartado 4.2 se muestra el ajuste del modelo logit multinomial. El apartado 4.3 presenta la construcción de los árboles de decisión. En estos dos últimos apartados se tiene en cuenta que inicialmente se construye el modelo y los árboles de decisión con todos los datos y las variables, se realiza una interpretación de los principales resultados y se verifican supuestos básicos, posteriormente se toma una muestra de los datos y se dividen en dos conjuntos; de entrenamiento y de prueba, adicionalmente se trabaja con reducción de variables (resultado del análisis preliminar), se revisa las tasas de clasificación errónea en ambas técnicas. Finalmente, en la sección 4.4 se efectúa una comparación de los resultados en el modelo logístico multinomial y los árboles de decisión.

En el capítulo de conclusiones se exponen los principales hallazgos en la comparación de estas dos técnicas. Finalmente, se muestra la bibliografía utilizada y en anexos el código en R de las rutinas implementadas.

3. Marco Teórico

A continuación, se presentan los principales fundamentos de los modelos logísticos y los árboles de decisión en el análisis de un conjunto de datos con variable respuesta del tipo categórica, en primer lugar, se habla de la regresión logística y posteriormente de los árboles de decisión y su implementación en R.

3.1 Regresión logística

La construcción de modelos estadísticos generalmente surge de la necesidad de cuantificar relaciones entre variables; la variable a ser explicada (respuesta) y las variables explicativas (regresoras). Dichos modelos son representados por ecuaciones matemáticas que dan cuenta de la relación entre las variables de interés para el investigador.

En el modelo de regresión lineal con variable respuesta continua, se deben cumplir supuestos básicos para iniciar en el tratamiento de los datos (variable respuesta con distribución normal, independencia entre las observaciones y homocedásticidad).

En algunas ocasiones, para garantizar el cumplimiento de los supuestos del análisis clásico se aplica una transformación a la variable respuesta. Sin embargo, una transformación aplicada a la variable respuesta no es garantía para conseguir las propiedades deseables. Adicionalmente, las predicciones deben ser hechas en la escala original de las variables, razón por la cual se requiere de una transformación inversa después de que el modelo ha sido construido y definido. Montgomery y Peck (1992) muestran como la transformación inversa de la variable usualmente no produce estimaciones insesgadas de la media.

Cuando la variable respuesta es discreta o categórica, el modelo lineal clásico no es apropiado. Nelder y Wedderburn (1972) extendieron la teoría de los modelos lineales a una familia más amplia. La familia exponencial de densidades, denominándolos Modelos Lineales Generalizados (GLM), en este

marco es donde ingresa el modelo de regresión logística, puesto que, este tipo de modelos maneja variable respuesta de tipo categórico (binario o multinomial).

En este trabajo se construye un modelo de regresión logística para respuesta multinomial, por tanto, inicialmente se especifica el modelo de regresión logística simple, sin entrar en muchos detalles y posteriormente se desarrolla de una manera más detallada la teoría relacionada con la regresión logística multinomial.

3.1.1 Regresión logística simple

La Regresión Logística Simple, desarrollada por David Cox en 1958, es un método de regresión que permite estimar la probabilidad de una variable cualitativa en función de una o varias variables cuantitativas o cualitativas.

Con la regresión logística se busca expresar la probabilidad de que ocurra un evento de interés, como función de algunas variables (covariables), que desde la teoría o la experiencia del investigador puedan ser influyentes en la respuesta.

Al considerar su forma más simple, el modelo de regresión logística para una variable aleatoria binaria Y es un modelo lineal para el logaritmo de la ventaja de respuesta $Y = 1$ en cada valor observado x de la variable explicativa, por tanto, queda especificado de la siguiente forma.

$$ln\left[\frac{p(x)}{1-p(x)}\right] = \alpha + \beta x$$

Adicionalmente, se puede expresar de la siguiente forma en términos de la probabilidad de respuesta 1 en x.

$$p(x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$

Donde; α es el valor común del logaritmo de las ventajas de respuesta $Y = 1$ frente a la respuesta $Y = 0$ cuando $\beta = 0$, es decir, cuando la respuesta es independiente de la variable explicativa; y β corresponde al efecto de la variable explicativa.

Adicionalmente, se tiene que la curva de probabilidad $p(x)$ es estrictamente creciente si $\beta > 0$ y estrictamente decreciente para $\beta < 0$.

En el caso general, que involucra *p-variables* explicativas $X_1, X_2, \dots, X_p$, asumiendo que el intercepto α se reemplace por β_0, el modelo queda definido de la siguiente forma:

$$p(x) = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_p x_p}}{1 + e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_p x_p}}$$

donde $x = (x_1, x_2, \dots, x_p)$, lo cual se conoce como función logística multivariada. Las variables explicativas pueden ser de tipo nominal, ordinal o continuo lo cual hace muy versátil la regresión logística.

3.1.2 Interpretación de parámetros

1. Si $\beta_i's = 0$ para $i \neq 0$ en el modelo multivariante, se tiene que: $p(x) = e^{\alpha}/(1 + e^{\alpha})$, lo que indica que la variable Y es independiente de X puesto que $p(x)$ no depende de x.

2. β_0 es el valor común del logaritmo de las ventajas de respuesta $Y = 1$ frente a la respuesta $Y = 0$ cuando $\beta_i's = 0$ para para $i \neq 0$

3. La fórmula general del modelo logit simple indica que, por cada unidad de incremento en X, el logit de respuesta 1 aumenta aditivamente en β unidades.

La interpretación adecuada de los coeficientes β en un modelo de regresión logística está estrechamente relacionado con los conceptos de: riesgo relativo, odds (ventaja) y razón de odds.

En este sentido, si se considera las $p - variables$ que conforman el vector X, es decir, $X = (X_1, X_2, \dots, X_P)'$, se puede probar por las propiedades de la función exponencial, que la ventaja de respuesta $Y = 1$ para dos combinaciones de valores de las variables explicativas $x_1 = (1, x_{11}, \dots, x_{1P})'$ y $x_2 = (1, x_{21}, \dots, x_{2P})'$ se puede escribir como sigue:

$$\theta(x_1, x_2) = \frac{\frac{p(x_1)}{1 - p(x_1)}}{\frac{p(x_2)}{1 - p(x_2)}} = \frac{exp\left(\sum_{r=0}^{R} \beta_p x_{1p}\right)}{exp\left(\sum_{r=0}^{P} \beta_p x_{2p}\right)} = exp\left(\sum_{r=1}^{R} \beta_p (x_{1p} - x_{2p})\right)$$

Para más detalles de la estimación de parámetros, medidas de bondad de ajuste y demás medidas asociadas a la regresión logística se puede consultar en: Agresti (1996), Díaz y Morales (2009), Agresti (2000), Aguilera y Escabias (2021).

3.1.3 Supuestos del modelo

Los modelos de regresión logísticos deben cumplir los siguientes supuestos básicos para su formulación:

- ✓ La variable respuesta (dependiente) es binaria.

- ✓ Independencia: La regresión logística supone que las observaciones del conjunto de datos son independientes entre sí. Es decir, las observaciones no deben provenir de mediciones repetidas de un mismo individuo ni estar relacionadas entre sí de ninguna manera.

- ✓ Linealidad: Uno de los supuestos más importantes en regresión logística es que la relación entre el logit o log-odds de la variable respuesta y cada variable predictora o variable independiente es lineal; este supuesto se

verifica únicamente para las variables numéricas continuas que se tengan en el modelo. La forma más fácil de evaluar esta suposición es usando la prueba de Box-Tidwell (1962).

3.2 Regresión logística multinomial

La regresión logística multinomial se utiliza cuando la variable dependiente o respuesta es del tipo cualitativa y posee más de dos categorías, por tanto, es una generalización de la regresión logística simple. Algunos ejemplos donde se puede emplear la regresión logística multinomial podrían ser:

- ✓ Estado civil (soltero, casado, viudo, divorciado)
- ✓ Elección de carrera profesional (Ingeniería, Educación, Salud, …)
- ✓ Tarea a realizar por un autor en un artículo (Análisis, concepción, desarrollo o escritura)

En este trabajo se usa la generalización de los modelos logit para una variable respuesta categórica politómica nominal.

3.2.1 Introducción

Sea Y una variable politómica nominal con más de dos categorías de respuesta $Y_1, Y_2, \dots, Y_S$, al igual que en la regresión logística de respuesta binaria, el objetivo es explicar la probabilidad de cada categoría de respuesta en términos de un conjunto de covariables $X_1, X_2, \dots, X_R$, de modo que el modelo ajustado sea de la forma:

$$p_s(x) = f_s(x) \quad \forall s = 1,2, \dots, S$$

donde; $p_s(x) = P[Y = Y_S \,/\, X = x]$, para cada vector x de valores observados de las variables explicativas y teniendo en cuenta que como la variable respuesta es politómica la distribución condicionada es una multinomial de parámetros las probabilidades de cada una de las categorías de respuesta.

Dado que en el modelo logit para respuesta binaria se ha construido como un modelo lineal para el logaritmo de la ventaja de respuesta $Y = 1$ frente a la respuesta $Y = 0$, para la respuesta politómica se puede definir una transformación logit para comparar cada par de categorías de respuesta. Para el caso se tienen un total de $\binom{S}{2}$ combinaciones que representan la cantidad de transformaciones logit de la forma.

$$ln\left[\frac{\frac{p_t(x)}{p_t(x)+p_s(x)}}{\frac{p_s(x)}{p_t(x)+p_s(x)}}\right] = ln\left[\frac{p_t(x)}{p_s(x)}\right], \quad \forall t, s = 1,2,\dots,S \;\; con \;\; t \neq s$$

La expresión anterior representa el logaritmo de la ventaja de respuesta Y_t frente a Y_s condicionado a las observaciones que caen en uno de ambos niveles, como es mencionado por Aguilera y Escabias (2021). El conjunto de transformaciones logit es redundante y para construir el modelo logit de respuesta multinomial bastará con considerar $(S-1)$ transformaciones logit básicas.

3.2.2 Regresión logística Multinomial Nominal

Cuando la variable de interés tiene S categorías de respuesta, el problema es la estimación de la probabilidad de pertenencia a cada una de las s-modalidades de la variable Y, para un individuo que tiene un perfil determinado por los valores que asuman las variables $X_1, X_2, \dots, X_R$. Díaz (2009), propone abordar este problema en base a transformaciones logit generalizadas definidas con respecto a una categoría o modalidad de referencia, generalmente se toma la última categoría como la de referencia, aunque en la práctica es posible definir

cuál es la que se tome como referencia. Las transformaciones logit generalizadas están definidas así:

$$L_s(x) = ln\left[\frac{p_s(x)}{p_S(x)}\right], \quad \forall s = 1,2,\dots,S-1$$

Donde $L_s(x)$ representa el logaritmo de la ventaja de la respuesta Y_s dado que la respuesta cae en la categoría Y_s o en la categoría Y_S. Adicionalmente cualquier transformación logit para un par de categorías se puede obtener a partir de sus transformaciones logit generalizadas asociadas en la forma.

$$ln\left[\frac{p_t(x)}{p_s(x)}\right] = L_t(x) - L_s(x) \quad \forall t, s$$

3.2.3 Formulación e interpretación

El modelo logit generalizado de respuesta nominal se formula como un modelo lineal para cada una de las transformaciones logit generalizadas, por tanto, para el caso de tener R variables explicativas $X_1, X_2, \dots, X_R$ el modelo queda especificado de la forma

$$L_s(x) = \sum_{r=0}^{R} \beta_{rs} x_r = x'\beta_s \quad \forall s = 1,2,\dots,S-1$$

Para cada vector de valores observados de las variables explicativas $x = (x_0, x_1, \dots, x_R)'$ con $x_0 = 1$ y $\beta_s = (\beta_{0s}, \beta_{1s}, \dots, \beta_{Rs})'$ el vector de parámetros asociado a la categoría Y_s.

Análogamente al modelo logit simple, el modelo logit de respuesta politómica se puede escribir en términos de las probabilidades de respuesta de la siguiente forma.

$$p_s(x) = \frac{exp(\sum_{r=0}^{R} \beta_{rs} x_r)}{1 + \sum_{s=1}^{S-1} exp(\sum_{r=0}^{R} \beta_{rs} x_r)} \quad \forall s = 1,2,\dots,S-1$$

$$p_S(x) = \frac{1}{1 + \sum_{s=1}^{S-1} exp(\sum_{r=0}^{R} \beta_{rs} x_r)}$$

Con el propósito de ilustrar, suponga que se tiene una variable respuesta con $S = 3$ modalidades y $R = 4$ variables explicativas. Lo anterior implica que se deben estimar dos conjuntos de 5 parámetros cada uno:

$$\begin{matrix} \alpha_1 & \beta_{11} & \beta_{12} & \beta_{13} & \beta_{14} \\ \alpha_2 & \beta_{21} & \beta_{22} & \beta_{23} & \beta_{24} \end{matrix}$$

así la probabilidad de que un sujeto este en la primera modalidad, se calcula a través de la expresión:

$$P(Y = 1) = \frac{exp[\beta_1{}'X]}{1 + exp[\beta_1{}'X] + exp[\beta_2{}'X]}$$

Para calcular la probabilidad de que el sujeto se encuentre en la segunda modalidad se emplea la fórmula.

$$P(Y = 2) = \frac{exp[\beta_2{}'X]}{1 + exp[\beta_1{}'X] + exp[\beta_2{}'X]}$$

donde;

$$exp[\beta_1{}'X] = exp[\alpha_1 + \beta_{11}X_1 + \beta_{12}X_2 + \beta_{13}X_3 + \beta_{14}X_4]$$
$$exp[\beta_2{}'X] = exp[\alpha_2 + \beta_{21}X_1 + \beta_{22}X_2 + \beta_{23}X_3 + \beta_{24}X_4]$$

y la probabilidad de que el sujete este en la tercera modalidad, se calcula por medio del complemento de los dos eventos previamente calculados, así:

$$P(Y = 3) = 1 - P(Y = 1) - P(Y = 2)$$

aquí se toma la tercera categoría como la categoría de referencia.

En el caso de una única variable explicativa cuantitativa X, el modelo es de la forma

$$L_s(x) = \alpha_s + \beta_s x. \quad \forall s = 1,2, \dots, S-1$$

Para cada valor observado x de la variable explicativa X, la exponencial de los parámetros β_s asociados a cada categoría de respuesta se interpretan en términos de cocientes de ventajas como sigue:

$$\theta_s(\Delta X = 1) = \frac{\frac{p_s(x+1)}{p_S(x+1)}}{\frac{p_s(x)}{p_S(x)}} = \frac{\exp\big(\alpha_s + \beta_s(x+1)\big)}{\exp(\alpha_s + \beta_s x)}$$

$$= \exp(\beta_s) \quad \forall s = 1,2,\dots,S-1$$

Siendo $\theta_s(\Delta X = 1)$ el cociente de ventajas de respuesta Y_s frente a la última Y_S cuando se incrementa en una unidad la variable X. En el modelo logit generalizado multinomial los cocientes de ventajas se definen incrementando una de las variables y controlando fijas las demás, por lo tanto, este cociente de ventajas queda definido como sigue:

$$\theta_s(\Delta X_l = 1 \,/\, X_r = x_r, r \neq l\,) = \frac{\frac{P(Y = Y_s \,/\, X_l = x_l + 1, X_r = x_r, r \neq l)}{P(Y = Y_S \,/\, X_l = x_l + 1, X_r = x_r, r \neq l)}}{\frac{P(Y = Y_s \,/\, X_l = x_l, X_r = x_r, r \neq l)}{P(Y = Y_S \,/\, X_l = x_l, X_r = x_r, r \neq l)}}$$

$$= \exp(\beta_{ls}) \qquad \forall s = 1,2,\dots,S-1$$

Donde $\theta_s(\Delta X_l = 1 \,/\, X_r = x_r, r \neq l\,)$ es el cociente de ventajas de respuesta Y_s frente a la última Y_S cuando se incrementa en una unidad la variable X_l y las demás se dejan fijas.

En muchas situaciones se puede dar el caso de que alguna de las variables explicativas sea categórica y deba ser incluida en el modelo, para lo cual esta es incorporada por medio de variables de diseño asociadas (Dummy). Tal como se muestra en Aguilera y Escabias (2021).

3.2.4 Estimación por máxima verosimilitud

Suponga que se dispone de una muestra aleatoria de tamaño N con Q combinaciones diferentes de valores de las variables explicativas $X_1, X_2, \dots, X_R$.

En cada combinación de los valores de las variables explicativas, denotadas por $x_q = \left(x_{q0}, x_{q1}, \dots, x_{qR}\right)'$ con $x_{q0} = 1$; $\forall q = 1, \dots, Q$, se dispone de una muestra aleatoria de n_q observaciones independientes de la variable de respuesta politómica Y, de entre las cuales se denotará por $y_{s/q}$ al número de observaciones que caen en la categoría de respuesta Y_s; $\forall s = 1, \dots, S$. Por lo tanto, se verifica que $\sum_{s=1}^{S} y_{s/q} = n_q$ y $\sum_{q=1}^{Q} n_q = N$.

Entonces los vectores $(y_{1/q}, \dots, y_{S/q})'$; $\forall q = 1, \dots, Q$ tienen distribución de probabilidad multinomiales independientes, $M(n_q; P_{1/q}, \dots, P_{S/q})$ verificando que $\sum_{s=1}^{S} P_{s/q} = 1$, donde $P_{s/q} = P\left[Y = Y_s / X = x_q\right]$.

La función de verosimilitud de los datos queda definida por la expresión:

$$\prod_{q=1}^{Q}\left(\frac{n_q!}{\prod_{s=1}^{S}(y_{s/q})!}\prod_{s=1}^{S} {P_{s/q}}^{y_{s/q}}\right),$$

de modo que el núcleo de la log-verosimilitud esta dado por

$$K = \sum_{q=1}^{Q}\sum_{s=1}^{S} y_{s/q} \ln\left(p_{s/q}\right)$$

Derivando respecto a los parámetros como sigue

$$\frac{\Delta K}{\beta_{rs}} = \sum_{q=1}^{Q} y_{sq} x_{qr} - \sum_{q=1}^{Q} n_q x_{qr} \frac{exp\left(\sum_{r=0}^{R} \beta_{rs} x_{qr}\right)}{\sum_{s=1}^{S} exp\left(\sum_{r=0}^{R} \beta_{rs} x_{qr}\right)}$$

Se obtienen las ecuaciones de verosimilitud cuya forma matricial es

$$X'_{((R+1)\times Q)} y_{s_{(Q\times 1)}} = X'_{((R+1)\times Q)} \widehat{m}_{s_{(Q\times 1)}};\ \forall s = 1,2, \dots, S-1,$$

siendo

$$y_s = (y_{s/1}, \dots, y_{s/Q})'$$
$$\widehat{m}_s = (\widehat{m}_{s/1}, \dots, \widehat{m}_{s/Q})'$$

$\widehat{m}_{s/q}$ corresponde a la frecuencia esperada de respuesta Y_s en la combinación x_q de valores observados de las variables explicativas, estimada bajo el modelo y definida por $\widehat{m}_{s/q} = n_q \hat{p}_{s/q}$.

La obtención de los estimadores de máxima verosimilitud de los parámetros se logra por la resolución simultanea de $(S-1)$ sistemas de $(R+1)$ ecuaciones no lineales cada uno. Por tanto, la solución se obtiene recurriendo a los métodos iterativos de Newton-Raphson.

El estimador de los parámetros $\hat{\beta}$ es un vector columna de dimensión $(S-1)$ definido por:

$$\hat{\beta} = \left(\hat{\beta}_1', \ldots, \hat{\beta}_{S-1}'\right)'$$

Para más detalles de la estimación, matriz de covarianzas y matriz de información de Fisher consulte en Aguilera y Escabias (2021).

3.2.5 Paquete nnet para ajuste de modelos logísticos multinomiales.

El paquete nnet de R ha sido usado ampliamente en la construcción de redes neuronales, sin embargo, se adapta a construcción de modelos logarítmicos lineales a través de redes neuronales, Venables y Ripley (2002).

Para ajustar un modelo se utiliza la función `multinom()`, a continuación se presenta la función y sus argumentos.

```
multinom(formula, data, weights, subset, na.action,
         contrasts = NULL, Hess = FALSE, summ = 0,
         censored = FALSE, model = FALSE, ...)
```

la especificación de los argumentos usados se presenta a continuación.

`formula`	Fórmula que define el modelo de regresión. La respuesta debe ser un factor o una matriz con K columnas, que se interpretarán como recuentos para cada una de las k clases.
`Data`	Un marco de datos (data.frame, entre otros)
`weights`	Pesos opcionales para el ajuste del modelo.
`subset`	Expresión que indica qué subconjunto de las filas de los datos debe usarse en el ajuste. Todas las observaciones se incluyen de forma predeterminada
`na.action`	Una función para filtrar los datos faltantes del marco del modelo.
`contrats`	Lista de contrastes que se utilizarán para algunos o todos los factores que aparecen como variables en la fórmula del modelo.
`Hess`	Valor lógico para indicar si el hessiano (la matriz de información observada/esperada) debe ser calculada.
`summ`	Entero; si no es cero, resume eliminando filas duplicadas y ajusta los pesos. Los métodos 1 y 2 difieren en velocidad (2 usos); El método 3 también combina filas con la misma X y diferente Y, lo que cambia la línea de base para la desviación.
`censored`	Si Y es una matriz con columnas, interpreta las entradas como una para las clases posibles, cero para las clases imposibles, en lugar de interpretarlos como recuentos.
`model`	Valor lógico. Si es true, el marco del modelo se guarda como componente del objeto devuelto.

Suponga que se asigna el nombre `mod.mult` a un objeto construido con la función `multinom()`, a partir de este objeto se puede extraer información importante para el análisis e interpretación del modelo, a continuación se especifica las funciones y su objetivo.

La función **`summary`**`(mod.mult)` muestra un resumen del modelo, que incluye los parámetros y errores entándar, de manera general muestra los logaritmos de las ventajas de la categoría Y_s frente a la categoría tomada como referencia. Si se desea determinar únicamente los coeficientes del modelo se puede usar la función `summary()` de la siguiente forma: **`summary`**`(mod.mult)$coefficients` y si el interés radica en los errores estándar

de los coeficientes se usa: `summary(mod.mult)$standard.errors`, estos últimos resultados se usan para determinar los valores experimentales del test de Wald.

Adicionalmente, los intervalos de confianza para los parámetros del modelo se pueden obtener con la función `confint(mod.mult)`. Las probabilidades para cada una de las categorías por individuo se pueden obtener con la función `predict(mod.mult,type="probs")`, mientras que la predicción de la categoría para cada uno de los individuos de la variable respuesta se puede calcular con `predict(ModTar,type="class")`.

Para mayor detalle sobre estos argumentos, revisar Venables y Ripley (2002). Y paquetes relacionados con modelación estadística.

3.3 Árboles de decisión

3.3.1 Introducción

Los árboles de decisión son un tipo de algoritmo de aprendizaje supervisado que se puede utilizar tanto en problemas de regresión como de clasificación (Breiman et al., 1984). Los creadores de la metodología del árbol de clasificación con aplicación en el aprendizaje autónomo, también conocida como metodología CART, fueron Leo Breiman, Jerome Friedman, Richard Olshen y Charles Stone, (Wu et al., 2008)

La aplicación en los terrenos de la estadística se inició en 1984. Esta metodología funciona tanto para variables de entrada y salida categóricas como continúas y clasifican grandes volúmenes de datos, de forma que se obtenga como resultado un modelo en forma de árbol. Estos están ubicados en el contexto de los modelos de predicción en el ámbito de la inteligencia artificial, (Breiman et al., 1984).

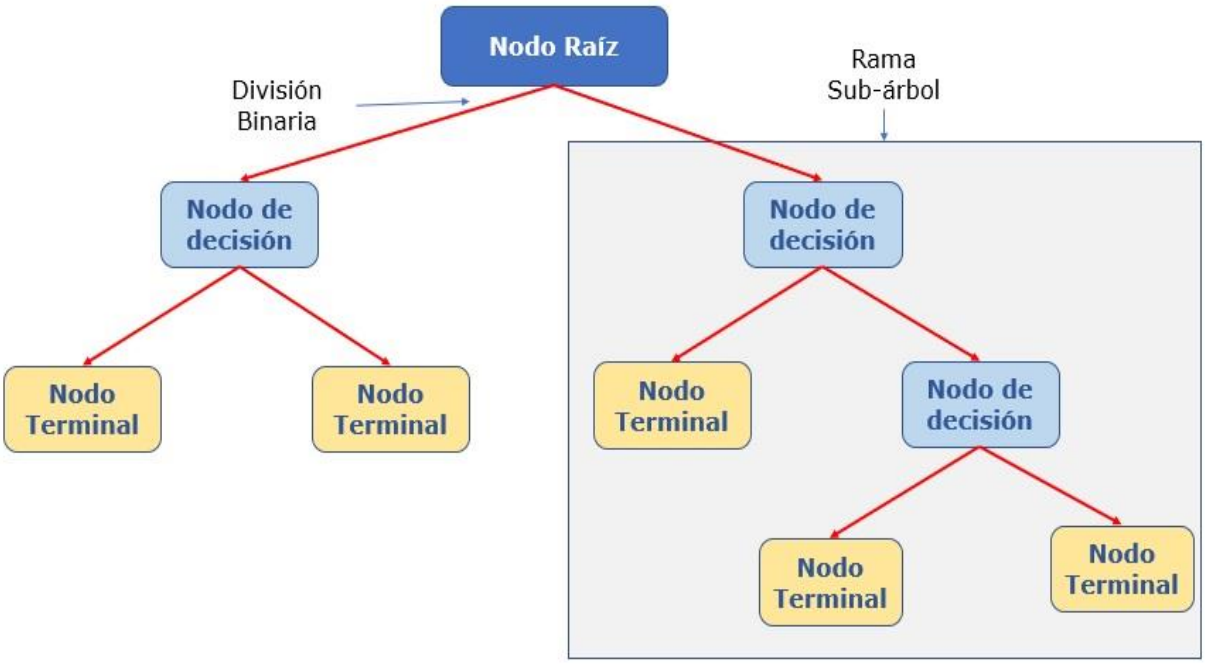
Como un árbol de decisión es un modelo de predicción, cuando se tiene un conjunto de datos, se construyen diagramas que siguen reglas lógicas que sirven

para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para la solución de un problema en particular.

Los modelos de árboles de decisión son comúnmente usados en la minería de datos para examinar los datos e inducir las reglas para realizar predicciones. Estos crecen a través de una división iterativa de grupos discretos donde el objetivo es maximizar la "distancia entre los grupos en cada división", Choque A. (2009). En la siguiente figura se encuentran algunos de los términos básicos asociados a los árboles de decisión.

Figura 1

Árbol de decisión y componentes



Nota: *Construcción del autor*

- **Nodo raíz:** Representa toda la población o muestra. Además, se divide en dos o más conjuntos homogéneos (División binaria)

- **División Binaria:** es un proceso de división de un nodo en dos o más subnodos, de acuerdo con una regla.

- **Nodo de Decisión:** Cuando un subnodo se divide en otros subnodos, se denomina nodo de decisión.

- **Nodo terminal:** Los nodos que no se dividen se denominan nodo terminal u hoja.

- **Poda:** Cuando se quitan los subnodos de un nodo de decisión, este proceso se denomina poda. Lo opuesto a la poda es la división.

- **Rama:** Una subsección de un árbol entero se llama Rama.

Un nodo, que se divide en subnodos, se denomina nodo padre de los subnodos. Mientras que los subnodos se denominan hijos del nodo padre**.**

3.2.2 Clasificación de los árboles de decisión:

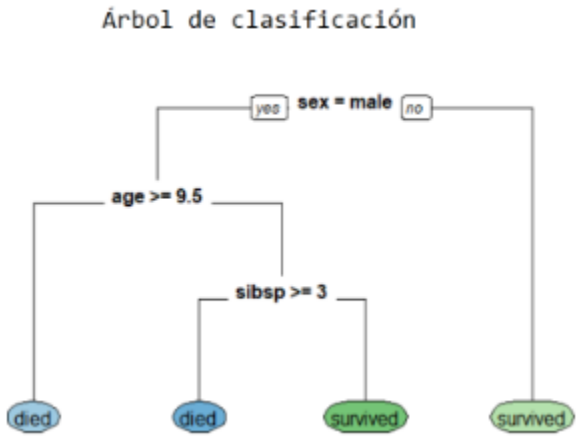
Dependiendo del tipo de la variable respuesta, los árboles de decisión se clasifican en: Árboles de clasificación o Árboles de regresión.

3.3.2.1 Árboles de clasificación.

Un árbol de decisión es de clasificación cuando la variable dependiente es de tipo cualitativo, una vez realizada las bifurcaciones, que siguen ciertas reglas el valor en el nodo terminal se reduce a la moda de las observaciones del conjunto de entrenamiento que han "caído" en esa región (James et al., 2013).

Figura 2

Árbol de clasificación



Nota: Tomado de https://fhernanb.github.io/libro_mod_pred/arb-de-regre.html

La creación de un árbol de decisión en el entorno de clasificación se lleva a cabo aplicando el algoritmo de Hunt, que se basa en la división en subconjuntos que buscan una separación óptima.

Una vez considerado un conjunto de datos para el entrenamiento de un nodo, si pertenecen a la misma clase se considera un nodo terminal, pero si pertenecen a varias clases, se dividen los datos en subconjuntos más pequeños en función de una variable y se repite el proceso. Para seleccionar qué variable elegir para obtener la mejor división se puede considerar el Error de Clasificación, el índice Gini (rpart) o la Entropía (C50).

Tasa de error de clasificación: Se puede definir la tasa de error de clasificación como la proporción con la que el algoritmo se equivoca. El error viene dado por:

$$E = \frac{Prediciciones\ incorrectas}{Total\ de\ predicciones}$$

Índice de Gini: El índice de Gini es una métrica de error alternativa que está diseñada para mostrar cuán "pura" es una región. La pureza en este caso

hace referencia a cuántos de los datos de entrenamiento en una región en particular pertenece a una sola clase. Si una región R_m contiene datos que son en su mayoría de una sola clase c, entonces el valor del índice de Gini será cercano a uno:

$$Gini = \sum_{c=1}^{C} (\hat{\pi}_{mc})^2$$

Donde; $\hat{\pi}_{mc}$ representa la fracción de datos usados en el entrenamiento de la región R_m que pertenecen a la clase c. A modo de ejemplo considere que se seleccionan dos datos de una región, entonces si la región es pura, estos deben ser de la misma clase y la probabilidad de que esto ocurra es 1, por tanto, una versión alternativa para el cálculo del índice de Gini está dada por la siguiente expresión.

$$Gini = 1 - \sum_{c=1}^{C} (\hat{\pi}_{mc})^2$$

Con esta última expresión se tiene que; índices Gini iguales a cero evidencia nodos puros (los datos pertenecen a una sola categoría), mientras que índices mayores a cero indican nodos impuros (los datos pertenecen a más de una categoría), en la medida que el valor se acerque a uno, los nodos presentan mayor impureza.

Entropía: Una tercera alternativa, que es similar en su concepción al índice de Gini, se conoce como entropía cruzada o desviación, La entropía se utiliza para calcular la homogeneidad de los datos en un nodo. Si los datos son completamente homogéneos (todos pertenecen a una misma categoría) su entropía es cero, por el contrario, si los datos pertenecen a más de una categoría, la entropía es mayor a cero. Por tanto, en la medida que el nodo es más impuro, mayor será su entropía, la entropía se calcula con la siguiente expresión:

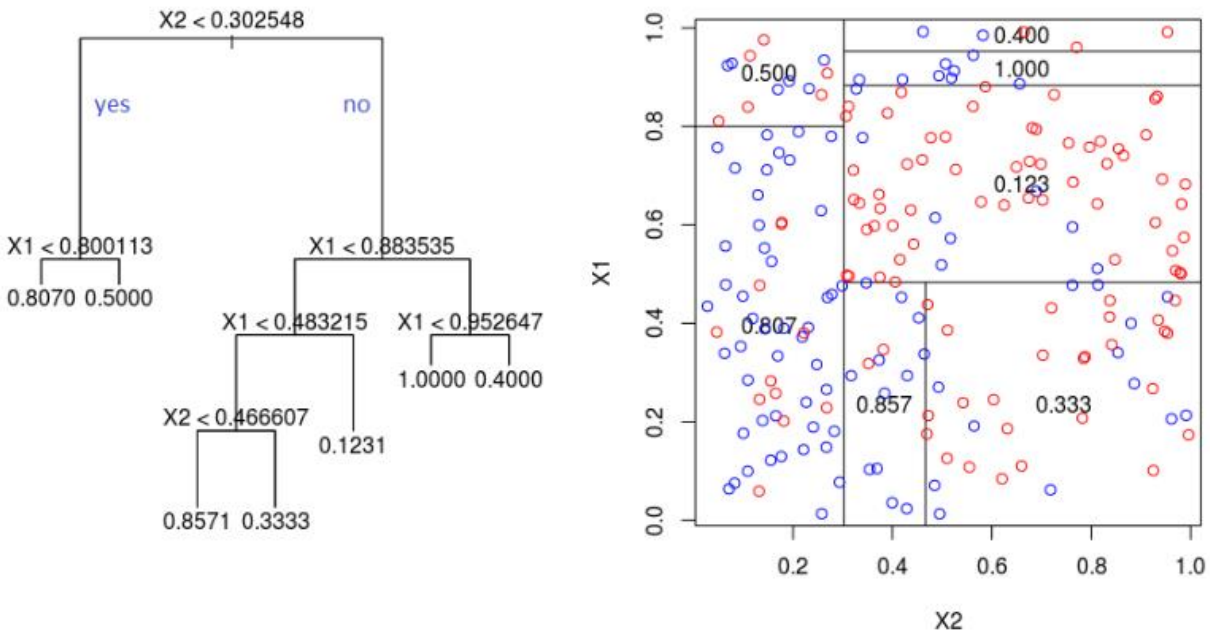
$$Entropía = -\sum_{c=1}^{C} \hat{\pi}_{mc} \log_2 \hat{\pi}_{mc}$$

3.3.2.2 Árbol de regresión.

Un árbol de decisión es de regresión cuando la variable dependiente es de tipo cuantitativo, la construcción de un árbol de regresión consiste en hacer preguntas del tipo ¿$x_k < c$?, para cada una de las covariables, de esta forma el espacio de las covariables es divido en hiperrectángulos y todas las observaciones que queden dentro de un hiperrectángulo tendrán el mismo valor estimado $\hat{y}$. (James et al., 2013).

Figura 3

Árbol de regresión



Nota: tomado de https://fhernanb.github.io/libro_mod_pred/arb-de-regre.html

En estos árboles de regresión la partición del espacio se hace de manera repetitiva para encontrar las variables y los valores de corte c de tal manera que se minimice la función de costos $\sum_{i=1}^{n}(y_i - \hat{y})^2$, lo cual indica que la suma residual de cuadrados se utiliza como criterio para realizar las divisiones binarias.

3.3.3 Ventajas y desventajas de los árboles de decisión

- ***Ventajas.***

- ✓ Los árboles de decisión son de fácil entendimiento, las reglas de clasificación se muestran y permiten un análisis sencillo de los resultados.
- ✓ Son útiles en la exploración de datos: se puede identificar la importancia de las variables a partir de cientos de ellas.
- ✓ Los outliers y valores faltantes no influyen en el modelo (A un cierto grado)
- ✓ El tipo de datos no es una restricción, se pueden utilizar variables respuesta del tipo cualitativo como cuantitativo, igual en las variables predictoras.
- ✓ Es un método no paramétrico (no hay suposición acerca del espacio de distribución y la estructura del clasificador)

- ***Desventajas.***

- ✓ En muchos casos los árboles presentan sobreajuste.
- ✓ Al categorizar las variables continuas se puede incurrir en pérdida de información.
- ✓ Precisión: métodos como las máquinas de vector soporte o Support Vector Machines (SVM) y clasificadores tipo ensamblador a menudo tienen tasas de error 30% más bajas que CART (Classification and Regression Trees).
- ✓ Inestabilidad: un pequeño cambio en los datos puede modificar ampliamente la estructura del árbol. Por lo tanto la interpretación no es tan directa como parece.

3.3.4 Paquetes de R para la construcción de árboles en R

Existen muchos paquetes en el entorno de R que permiten la construcción tanto de árboles de clasificación como de regresión. Los más conocidos para realizar la construcción de los árboles son:

- ✓ **tree**
- ✓ **rpart**
- ✓ **party**

En este trabajo solo se usa el paquete tree, por tanto, a continuación, se especifican las funciones y argumentos usados en la construcción de los árboles con este paquete.

Paquete Tree.

Un árbol se cultiva mediante particiones recursivas binarias utilizando la respuesta en la fórmula especificada y eligiendo divisiones de los términos del lado derecho (Ripley, 2019)

Este paquete propuesto por Brian Ripley en 2019, permite construir árboles de decisión y de clasificación, la función que permite la construcción de árboles es `tree()` y presenta la siguiente estructura.

```
tree(formula, data, weights, subset,
     na.action = na.pass, control = tree.control(nobs, ...),
     method = "recursive.partition",
     split = c("deviance", "gini"),
     model = FALSE, x = FALSE, y = TRUE, wts = TRUE, ...)
```

La especificación de los argumentos usados se presenta a continuación.

`formula`	Una expresión que define el modelo. El lado izquierdo (var respuesta) debe ser un vector numérico cuando se ajuste un árbol de regresión, o un factor cuando se produce un árbol de clasificación. El lado derecho debe ser una serie de variables numéricas o de factores separadas por ; no debe haber términos de interacción y se permite el uso de +,-, offset
`Data`	Un marco de datos (data.frame, entre otros)

`weights`	Vector de pesos observacionales no negativos; se permiten pesos fraccionarios.
`subset`	Expresión que especifica el subconjunto de casos que se van a utilizar.
`na.action`	Una función para filtrar los datos faltantes del marco del modelo. El valor predeterminado es (no hacer nada) ya que maneja los valores faltantes (dejándolos caer por el árbol en la medida de lo posible)
`control`	Lista presentada por tree.control
`method`	Proporciona el método que se va a utilizar. El único otro valor útil es.`"model.frame"`
`split`	Criterio de división a utilizar (Deviance o Gini)
`model`	Si el argumento es lógico y verdadero, el marco del modelo se almacena como componente en el resultado.
`x`	lógico. Si es true, se muestra la matriz de variables para cada caso.
`y`	lógico. Si es true, se muestra la variable respuesta.
`wts`	lógico. Si es cierto, los pesos son mostrados.

La función `summary()` arroja el resumen del árbol de decisión construido. Este resumen da información relacionada con: las variables involucradas, el número de nodos terminales, la desviación media residual, así como la tasa de clasificación errónea.

La representación gráfica del árbol de decisión se realiza a través de la función `plot(obj.tree)`. En este caso `obj.tree` hace referencia a un objeto creado por la función `tree()`, es decir, el nombre del objeto creado para almacenar el árbol de decisión construido. Luego para agregar las etiquetas de texto se agrega la función `text(obj.tree,pretty=0)`, por consiguiente la estructura para crear la gráfica del árbol de decisión será la siguiente.

```
plot(obj.tree)
text(obj.tree,pretty=0)
```

Dependiendo de la estructura resultante del árbol construido, es posible que se quiera un resumen un poco más detallado del mismo. Para obtener un resumen detallado del árbol, simplemente se escribe en R el nombre dado al árbol por medio de la función `tree()`. Será útil si desea conocer detalles del árbol para otros fines.

Cuando el árbol tiene muchos nodos terminales, puede que la interpretación de éste no sea de lo más sencillo, por tanto, en ocasiones se recurre a podar el árbol. Para ello es necesario crear un conjunto de entrenamiento y uno de prueba dividiendo el marco de datos en una muestra de entrenamiento y una de prueba.

Considere que se ha nombrado por `entren` al conjunto seleccionado como datos de entrenamiento, para realizar la poda y construir el árbol nuevamente se ajusta el modelo con la función `tree()`, utilizando la misma fórmula, excepto que se indica en el argumento `subset=entren`, es decir, que use como subconjunto de entrenamiento los datos de la muestra de entrenamiento, las instrucciones para la construcción del árbol y su gráfica se muestran a continuación.

```
Obj.tree<-tree(formula, data, weights, subset=entren, … )
plot(obj.tree)
text(obj.tree,pretty=0)
```

Las predicciones en el conjunto de prueba se pueden obtener por medio de la función `predict()`, a continuación, se presenta la función y sus argumentos.

```
predict(object, newdata = list(),
        type = c("vector", "tree", "class", "where"),
        split = FALSE, nwts, eps = 1e-3, ...)
```

`object` modelo ajustado con `tree()`.

`newdata` marco de datos que contiene los valores a los que se requieren las predicciones

`type`	cadena de caracteres que indica si las predicciones se devuelven como un vector (predeterminado) o como un objeto de árbol.
`split`	Determina el trabajo con los valores faltantes.
`nwts`	pesos para los casos, utilizados al predecir un árbol.
`eps`	Fija un límite inferior para las probabilidades, utilizado si se producen eventos de probabilidad cero predicha al predecir un árbol.

Con estas predicciones la idea es ir revisando que tan bien se realiza esta clasificación y para ello se determina la tasa de clasificación, para lo que se puede usar la siguiente función.

```
with(Datos[-entren,], table(tree.pred, Var))
```

Donde el primer argumento hace referencia al conjunto de datos prueba y el segundo a las predicciones hechas sobre el mismo, que da como resultado una tabla en la que en las diagonales están las clasificaciones correctas, mientras que fuera de las diagonales están las incorrectas y a partir de estos valores se puede determinar la tasa de clasificación.

Cuando se cultiva un árbol y este es de gran tamaño, puede llegar a tener demasiada variación. Por lo tanto, se puede usar la validación cruzada para podar el árbol de manera óptima, usando la tasa de clasificación errónea como base para la poda, la función `cv.tree()` permite realiza la validación cruzada. A continuación, se presenta la función y sus argumentos.

```
cv.tree(object, rand, FUN = prune.tree, K = 10, ...)
```

`object`	Un objeto de clase."`tree`".
`rand`	Opcionalmente un vector entero de la longitud del número de casos utilizados, asignando los casos a diferentes grupos para la validación cruzada.
`FUN`	La función para hacer la poda.
`K`	El número de pliegues de la validación cruzada.

La salida de los resultados de la poda por validación cruzada muestra los detalles de la ruta de la validación cruzada, también se pueden ver los tamaños de los árboles a medida que se podaron, las desviaciones a medida que avanzaba la poda, así como el parámetro de complejidad de costos utilizado en el proceso. Si se quiere tener una representación gráfica de esto, se usa la función `plot()` de la siguiente forma.

```
Arb.val.cru = cv.tree(object, FUN = prune.misclass)
plot(Arb.val.cru)
```

En la gráfica se identifican los tamaños de los árboles y las tasas de error, se escoge el tamaño de árbol que genere la menor tasa de error. Este último valor determina el tamaño del árbol óptimo. Para implementar la optimización del árbol por poda se utiliza la función `prune.misclass()`. A continuación, se presenta la función y los argumentos usados.

```
prune.tree(tree, k = NULL, best = NULL, newdata, nwts,
           method = c("deviance", "misclass"), loss,
           eps = 1e-3)
```

`tree`	Un objeto de clase, "`tree`". Modelo ajustado para el árbol.
`k`	parámetro costo-complejidad que define un subárbol específico de (un escalar), si falta se define algorítmicamente.
`best`	entero que solicita el tamaño (es decir, el número de nodos terminales óptimo) de un subárbol específico en la secuencia costo-complejidad que se va a devolver.
`newdata`	marco de datos sobre el que se evalúa la secuencia de subárboles de costo-complejidad. Si faltan, se utilizan los datos utilizados para cultivar el árbol.
`method`	`method = "misclass"`, cadena de caracteres que denota la medida de la heterogeneidad del nodo utilizada para guiar la poda de costo-complejidad.

Los demás argumentos ya se han especificado en otras funciones. El nuevo árbol con el número de nodos terminales definido se puede graficar utilizando la función `plot()` y agregándole las etiquetas con `text()`. Finalmente, si se desea evaluar la tasa de error de clasificación, se usa la función `predict()` para determinar las predicciones y `with()` para construir la tabla, tal como se usó en un apartado anterior.

3.4 Matriz de Confusión

La matriz de confusión es una tabla que se utiliza para determinar el rendimiento de un modelo de clasificación en un conjunto de datos de prueba para los que se conocen los valores reales. Esta matriz se puede utilizar tanto en el contexto de la regresión logística como en los árboles de clasificación.

Una observación del conjunto de datos es clasificada correctamente por el modelo logístico o por el árbol de decisión cuando el valor real de la variable respuesta coincide con su valor estimado por el modelo. Consideremos la siguiente tabla de contingencia.

Figura 4

Matriz de confusión

		Predicción	
		Positivo	Negativo
Realidad	Positivo	Verdaderos positivos	Falsos positivos
	Negativo	Falsos negativos	Verdaderos negativos

De acuerdo con la tabla anterior, la capacidad predictiva del modelo construido se puede medir de acuerdo con los conceptos de sensibilidad y especificidad, como se muestra a continuación.

$$sensibilidad = \frac{VP}{VP + FN}$$

$$especificidad = \frac{VN}{VN + FP}$$

Por lo tanto, la probabilidad de predecir correctamente el "éxito" de la variable respuesta, se denomina sensibilidad y la probabilidad de predecir correctamente el "fracaso" de la variable respuesta se denomina especificidad.

Adicionalmente, se puede determinar la tasa de clasificación errónea, la cual de manera general define que tanto se equivoca el modelo construido. Para su cálculo se podría usar la siguiente expresión.

$$Clasificación\ errónea = \frac{VP + VN}{VP + FP + FN + VN}$$

Este último resultado se usa tanto en los modelos de regresión logística multinomial como en los árboles de decisión y de regresión. Para el modelo multinomial y árboles con variables categóricas, la tabla de clasificación es más grande. La tabla indicada anteriormente es para regresión logística binaria y árboles binarios.

4. Resultados

4.1 Análisis descriptivo

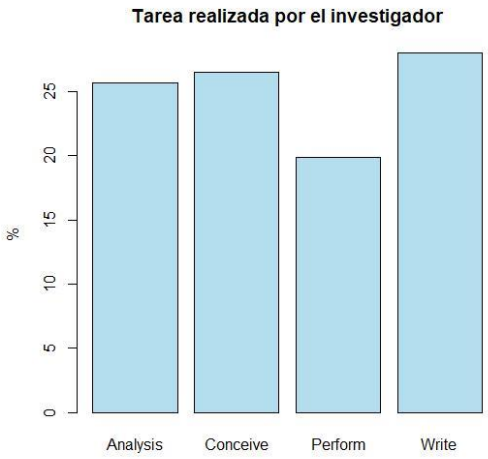
El conjunto de datos utilizados en esta investigación corresponde a registros de publicación de artículos científicos utilizados en una investigación realizada por Robinson-Garcia, Costas, Sugimoto, Larivière, y Nane en 2020. La base originalmente contaba con, 207841 filas, cada una de las cuales hace referencia a la participación de un autor en un artículo en particular. Sin embargo, esta se reduce a 207829 filas, puesto que en el análisis preliminar se identificó la presencia de datos atípicos.

La variable respuesta corresponde a Task y hace referencia a la tarea realizada por el investigador en el artículo en cuestión. Mientras las variables regresoras son; au_count: posición de la firma del investigador, n_authors: número de autores en el artículo, n_countries: número de países involucrados, n_institutes: número de instituciones involucradas, p_age: edad académica del autor en el año de publicación del artículo y pubs_age: número de artículos publicados por el autor hasta el año de publicación del artículo. A continuación, se presenta un análisis descriptivo de estas variables.

- **Task:** Variable de tipo cualitativa nominal que describe la tarea realizada por el investigador, La variable está dividida en cuatro posibles tareas a desarrollar por el investigador en la ejecución de la investigación científica que conlleva a la publicación de un artículo. Estas tareas son; Análisis (análisis de resultados), concebir (aporte de la idea original de la investigación), desarrollo (desarrollo de la investigación) y escritura (redacción del artículo). En la siguiente gráfica se puede apreciar la distribución de las distintas categorías.

Figura 5

Distribución por tarea



Nota: *Elaborada con los datos de la base*

De acuerdo con los resultados la tarea que predomina en los investigadores es la de Escritura con un 27.98%, sin embargo, no difiere en gran medida de las tareas de Análisis y concepción

- **au_count:** Posición de firma del investigador en el artículo en cuestión: variable de tipo ordinal, la distribución según la posición de la firma es:

Figura 6

Distribución de la posición de la firma en el artículo

1	2	3	4
42.01%	27.07%	18.97%	11.93%

Nota: *Elaborada con los datos de la base*

Lo anterior evidencia que predomina la primera posición en la firma del investigador en el artículo y a medida que la posición se ubica en lugares posteriores a la primera posición el porcentaje disminuye.

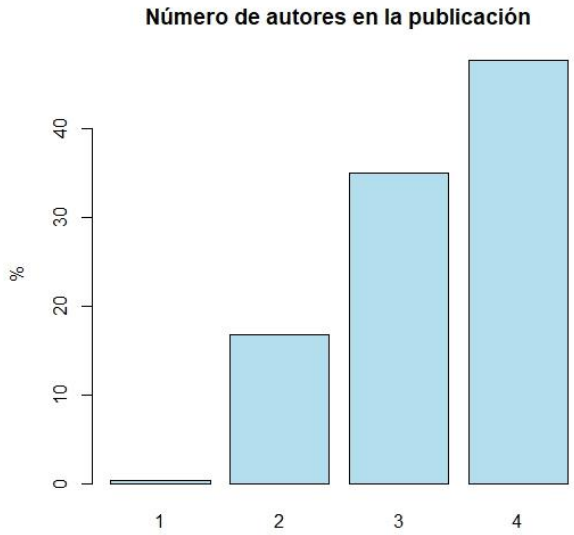
- **n_authors:** Número de autores en el artículo, se identifica que en estas investigaciones el número máximo de autores es de 4, siendo este el porcentaje mayor con un 47.74%. Adicional a esto se muestra que la publicación de artículos de manera individual es baja (0.37%) y que este porcentaje aumenta con la cantidad de autores.

A continuación, se presentan otras medidas de tipo descriptivo para el número de autores y la distribución por número de autores.

Mediana	Media	Moda	Desviación estándar	Coeficiente de variación
3	3.302	4	0.7546	22.85%

Figura 7

Distribución del número de autores



Nota: *Elaborada con los datos de la base de datos*

- **n_countries**: Número de países involucrados en el artículo. Se logra determinar que generalmente el artículo se realiza a nivel nacional, es decir, un solo país se involucra en la investigación (77.36% de los casos) y a medida que aumentan la cantidad de países involucrados el porcentaje disminuye, la mayor participación en cuanto a cantidad de países es de 6 con un 0.003%

A continuación, se presentan otras medidas de tipo descriptivo para el número de países y su distribución.

Mediana	Media	Moda	Desviación estándar	Coeficiente de variación
1	1.264	1	0.527779	41.74%

Figura 8

Distribución del número de países

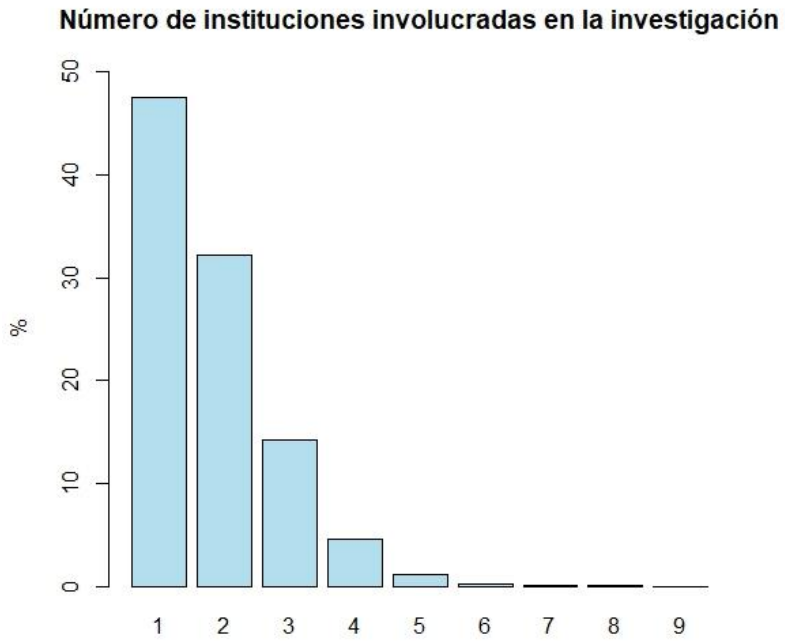
Nota: *Elaborada con los datos de la base de datos*

- **n_institutes:** Número de instituciones involucradas. Esta variable presenta un comportamiento similar a la anterior, es decir, a medida que aumenta la colaboración entre instituciones para la publicación de artículos, el porcentaje disminuye. Predomina la participación de una única institución (47.52%) y dos instituciones (32.24%), el máximo de

instituciones participantes en una misma investigación es de 9 con un 0.005%. A continuación, su distribución.

Figura 9

Distribución del número de instituciones

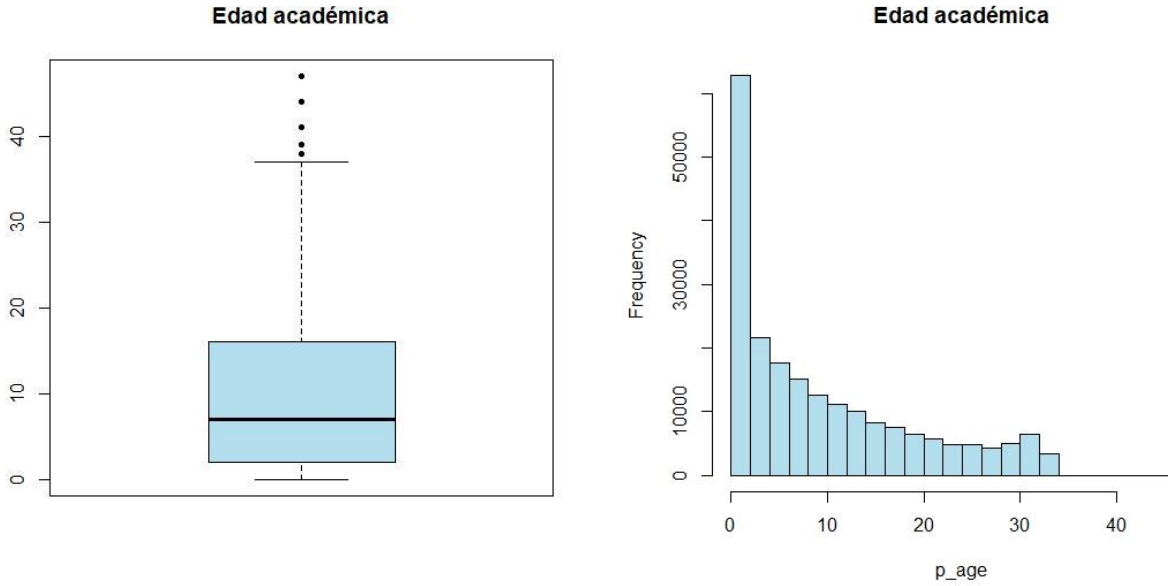


Nota: *Elaborada con los datos de la base de datos*

- **p_age:** Edad académica del autor en el año de publicación del artículo. Se aprecia que los primeros años de vida académica de los autores son los más productivos en cuanto a la publicación de artículos, el promedio de la edad académica de los autores está en 9.75 años, pero el 50% de los autores tiene una edad inferior a 7 años de edad académica. En esta variable se presentó el dato atípico, un autor con 99 años de vida académica, razón por la cual se excluyeron 3 artículos en los cuales participa. Así se eliminaron 12 filas de la base de datos, generalmente la firma de este autor ocupaba la cuarta posición en los artículos publicados. En la siguiente grafica se puede apreciar la distribución de la variable.

Figura 10

Distribución de la edad académica



Nota: *Elaborada con los datos de la base de datos*

- **pubs_age:** Número de artículos publicados por el autor hasta el año de publicación del artículo, el promedio de artículos publicados es de 40, el 50% de los autores tienen menos de 13 artículos publicados hasta el año de interés. A continuación, se presentan algunas medidas asociadas.

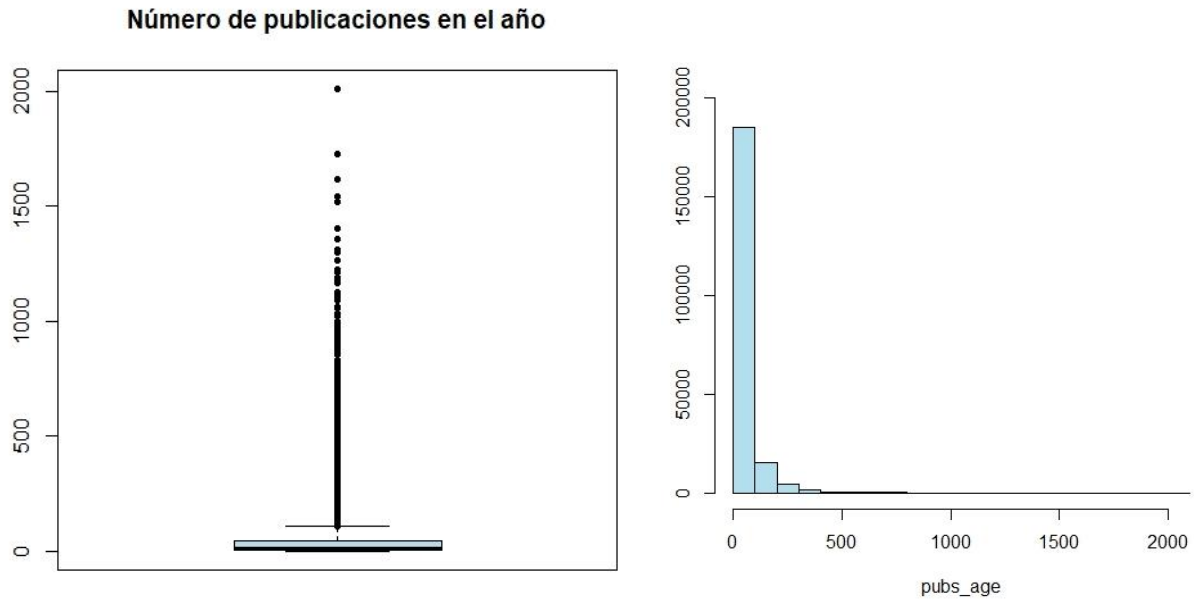
Mediana	Media	Moda	Desviación estándar	Coeficiente de variación
13	40.07	1	74.14	185%

Con los resultados anteriores se evidencia gran variabilidad en el número de artículos publicados por los investigadores involucrados en la base de datos,

se presenta la distribución del número de artículos publicados en el año de la publicación del artículo en cuestión.

Figura 11

Distribución por número de publicaciones



Nota: *Elaborados con los datos de la base de datos*

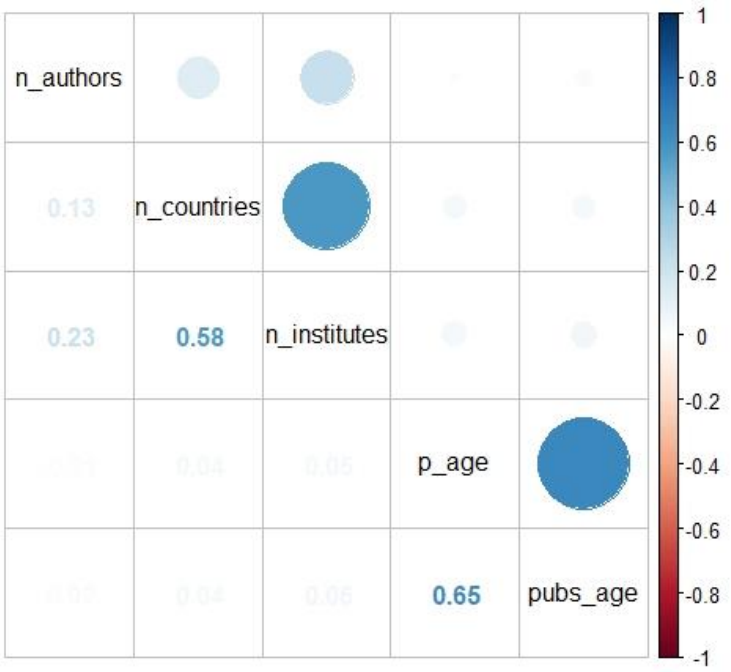
Como se puede evidenciar se presentan datos atípicos, que para el caso corresponden a autores que tienen un gran número de publicaciones en el año. Los datos que se analizan provienen del sector salud y allí un mismo individuo puede hacer múltiples análisis de datos y estos análisis hacen parte de investigaciones, por tanto, se presentan altos números de publicaciones de un autor.

- **Análisis de correlación entre las covariables del tipo numérico:** Las covariables presentan correlaciones de tipo positivo, especialmente las parejas de variable; Número de instituciones y de países involucrados en la investigación (0.58) y la edad académica y número de publicaciones en

el año de la publicación (0.65), estos resultados se pueden apreciar en la siguiente gráfica.

Figura 12

Matriz de correlaciones



Nota: *Elaborados con los datos de la base de datos*

Dado que el objetivo de la investigación corresponde a la predicción de la tarea realizada por el investigador en diferentes artículos científicos, basándose en las variables regresoras descritas anteriormente. A continuación, se presenta el análisis de los datos con las técnicas de Modelo Logit multinomial y Árboles de decisión.

4.2 Modelo Logit Multinomial

Inicialmente, se construye el modelo Logit para la respuesta. Tarea realizada por el investigador en función de las variables; posición de la firma,

número de autores, países, instituciones, edad académica y número de artículos publicados, a fin de seleccionar las variables significativas para explicar la respuesta, se usan todos los datos en la construcción del modelo. Posteriormente se construye el modelo con datos de entrenamiento y las variables que en el análisis previo arrojen ser significativas para el modelo.

4.2.1 Modelo Logit Multinomial sin datos de entrenamiento

El modelo propuesto queda especificado de la siguiente forma:

$$Task = au\_count + n\_authors + n\_countries + n\_institute + p\_age + pubs\_age$$

A continuación, se presentan los coeficientes resultantes del ajuste de un modelo logístico multinomial, que tiene por variable respuesta la tarea que realiza el investigador. Como categoría de referencia se presenta la tarea de Análisis.

Coefficients:

	(Intercept)	au_count2	au_count3	au_count4	n_authors
Conceive	0.04948619	-0.01872929	0.12440054	0.3034595	-0.09582012
Perform	-0.25184931	0.01516832	-0.18817588	-0.5648789	0.10267869
Write	0.08774832	-0.03650046	0.09326548	0.2302688	-0.10565501

	n_countries	n_institutes	p_age	pubs_age
Conceive	0.012034747	0.027018621	0.01440748	0.0006082762
Perform	-0.000176782	-0.004876688	-0.02425821	-0.0028405572
Write	0.059561167	0.045415577	0.01227314	0.0006401688

Std. Errors:

	(Intercept)	au_count2	au_count3	au_count4	n_authors	n_countries
Conceive	0.03235381	0.01567372	0.01852509	0.02307406	0.009139163	0.01427369
Perform	0.03461425	0.01600247	0.02075059	0.02931185	0.009636472	0.01572103
Write	0.03172279	0.01539162	0.01827254	0.02290463	0.008981584	0.01391270

	n_institutes	p_age	pubs_age
Conceive	0.007964535	0.0008866711	0.0001084685
Perform	0.008713025	0.0011965646	0.0002114321
Write	0.007819060	0.0008792312	0.0001079347

Residual Deviance: 561881.9
AIC: 561935.9

Dado que la función usada en el ajuste del modelo da las transformaciones logit generalizadas con respecto a la categoría de referencia, se tiene que;

$$L_s(x) = ln\left[\frac{p_s(x)}{p_0(x)}\right], \forall s = 1,2,3.$$

donde; $L_s(x)$ representa el logaritmo de la ventaja de respuesta Y_s respecto a la respuesta Y_0. Por lo tanto, el modelo queda especificado de la siguiente forma, teniendo en cuenta que las variables X_1, X_2 y X_3 son variables de diseño para la posición de la firma en el artículo:

$$L_s(x) = \beta_{s0} + \beta_{s1}X_1 + \beta_{s2}X_2 + \beta_{s3}X_3 + \beta_{s4}X_4 + \beta_{s5}X_5 + \beta_{s6}X_6 + \beta_{s7}X_7 + \beta_{s8}X_8$$
$$\forall s = 1,2,3$$

Así, para cada una de las categorías de la variable respuesta se tiene que:

$$L_1(x) = \beta_{10} + \beta_{11}X_1 + \beta_{12}X_2 + \beta_{13}X_3 + \beta_{14}X_4 + \beta_{15}X_5 + \beta_{16}X_6 + \beta_{17}X_7 + \beta_{18}X_8$$

$$L_2(x) = \beta_{20} + \beta_{21}X_1 + \beta_{22}X_2 + \beta_{23}X_3 + \beta_{24}X_4 + \beta_{25}X_5 + \beta_{26}X_6 + \beta_{27}X_7 + \beta_{28}X_8$$

$$L_3(x) = \beta_{30} + \beta_{31}X_1 + \beta_{32}X_2 + \beta_{33}X_3 + \beta_{34}X_4 + \beta_{35}X_5 + \beta_{36}X_6 + \beta_{37}X_7 + \beta_{38}X_8$$

donde, para cada una de las variables explicativas se tienen 3 parámetros (número de categorías de la respuesta menos uno), así por ejemplo; para X_1 (segunda posición en las firmas de los investigadores) se tienen los parámetros β_{11}, β_{21} y β_{31}. Así con las demás variables explicativas. También hay tres parámetros independientes, β_{10}, β_{20} y β_{30} correspondientes a los interceptos.

La exponencial de los parámetros estimados se pueden interpretar en términos de cocientes de ventajas y son presentados a continuación, junto con algunas interpretaciones.

	(Intercept)	au_count2	au_count3	au_count4	n_authors	n_countries
Conceive	1.0507311	0.9814450	**1.132469**	1.354537	0.9086274	1.0121075
Perform	0.7773619	1.0152839	**0.828469**	**0.568429**	1.1081353	0.9998232
Write	1.0917133	0.9641577	1.097753	1.258938	**0.8997350**	1.0613707

	n_institutes	p_age	pubs_age
Conceive	1.0273869	1.0145118	1.0006085
Perform	0.9951352	**0.9760337**	0.9971635

```
Write       1.0464627 1.0123488 1.0006404
```

- $\mathbf{exp}(\widehat{\boldsymbol{\beta}}_{\mathbf{12}}) = \mathbf{1.132469}$ es el cociente de ventajas de la tarea Concepción frente a la tarea Análisis, cuando un individuo presenta su firma en el tercer lugar del artículo en lugar del primero, manteniendo constante; el número de autores, países, institutos, años de vida académica y publicaciones del autor en el año de publicación del artículo. Esto indica que la ventaja de la tarea Concepción frente a la tarea Análisis de los autores que firman en tercer lugar es 1.132469 veces mayor que la misma ventaja de los autores que firman en primer lugar.

- $\mathbf{exp}(\widehat{\boldsymbol{\beta}}_{\mathbf{22}}) = \mathbf{0.828469}$ es el cociente de ventajas de la tarea Desarrollo frente a la tarea Análisis, cuando un individuo presenta su firma en el tercer lugar del artículo en lugar del primero, manteniendo constantes los valores de las otras variables. Esto indica que la ventaja de la tarea Desarrollo frente a la tarea Análisis de los autores que firman en tercer lugar es 0.828469 veces menor que la misma ventaja de los autores que firman en primer lugar. Equivalentemente la ventaja de la tarea Desarrollo frente a la tarea Análisis de los autores que firman en primer lugar es 1/0.828469 = 1,207045768 mayor que la misma ventaja de los autores que firman en tercer lugar.

- La ventaja de la tarea Desarrollo frente a la tarea Análisis de los autores que firman en primer lugar es 1/0.568429 = 1,7592347 mayor que la misma ventaja de los autores que firman en cuarto lugar, manteniendo constantes los valores de las otras variables, el resultado anterior se obtiene al tener en cuenta que $\mathbf{exp}(\widehat{\boldsymbol{\beta}}_{\mathbf{23}}) = \mathbf{0.568429}$

- $\mathbf{exp}(\widehat{\boldsymbol{\beta}}_{\mathbf{34}}) = \mathbf{0.8997350}$ es el cociente de ventajas de la tarea Escritura frente a la tarea Análisis, cuando en un artículo se incrementa en 1 la cantidad de autores, manteniendo constantes las demás variables. Es decir, que por cada autor adicional la ventaja de la tarea Escritua es 0.8997350 veces menor que la misma ventaja de la tarea Análisis.

- $\mathbf{exp}(\widehat{\boldsymbol{\beta}}_{27}) = \mathbf{0.9760337}$ es el cociente de ventajas de la tarea Desarrollo frente a la tarea Análisis, cuando en un artículo se incrementa en 1 año la vida académica del autor, manteniendo constantes las demás variables. Es decir, que por cada año adicional de vida académica del autor la ventaja de la tarea Desarrollo frente a la tarea Análisis casi no se modifica. Sin embargo, por cada 5 años de vida académica del autor la ventaja de la tarea Desarrollo frente a la tarea Análisis disminuye y se multiplica por $\mathbf{exp}(\mathbf{5} * \widehat{\boldsymbol{\beta}}_{27}) = \mathbf{0.9760337^5} = \mathbf{0.88577639}$

La interpretación de parámetros se ha realizado sin tener en cuenta la significancia de estos. A continuación, se realiza una selección stepwise para el modelo multinomial y al modelo resultante se le analiza de significancia de los parámetros resultantes.

La selección Stepwise parte del modelo que solo contempla el intercepto y a medida que la incorporación de variables regresoras va disminuyendo el valor del AIC, estas van ingresando al modelo. En este sentido el método stepwise selecciona las variables; edad académica del autor, posición de la firma del autor en el artículo, número de autores del artículo, número de artículos publicados por el autor hasta el año de publicación del artículo, número de instituciones involucradas y número de países involucrados en el artículo en cuestión, es decir, que todas las variables son predictoras de la tarea desempeñada por el autor. Sin embargo, algunas de estas variables hacen una pequeña reducción del AIC. A continuación, se presenta el orden de ingreso de las variables al modelo y el valor de reducción del AIC:

- **p_age:** Edad académica del autor, reducción del AIC (8928.8)
- **au_count:** Posición de la firma del autor en el artículo, reducción del AIC (1039.7)
- **n_authors:** Número de autores del artículo, reducción del AIC (519.7)
- **pubs_age:** Número de artículos publicador por el autor hasta el año de publicación, reducción del AIC (388.1)

- **n_institutes:** Número de instituciones involucradas, reducción del AIC (132.1)
- **n_countries:** Número de países, reducción del AIC (18.8)

De acuerdo con los resultados anteriores, las variables; Edad académica del autor y la posición de la firma son las que predicen de mejor manera la tarea realizada por el autor en la producción del artículo en cuestión. Los resultados arrojados en el análisis Stepwise se pueden encontrar en el Anexo B.

La función utilizada en R para el ajuste del modelo no muestra por defecto la significación de parámetros. Para ello, asumiendo la distribución normal asintótica de los parámetros, se pueden obtener los valores experimentales del test de Wald para cada parámetro, los resultados a continuación.

```
         (Intercept)  au_count2 au_count3 au_count4 n_authors n_countries
Conceive    1.529532 -1.1949488  6.715246  13.15154 -10.48456  0.84314220
Perform    -7.275886  0.9478736 -9.068461 -19.27135  10.65522 -0.01124494
Write       2.766097 -2.3714495  5.104135  10.05337 -11.76352  4.28106590

         n_institutes     p_age   pubs_age
Conceive    3.3923664  16.24895   5.607860
Perform    -0.5597009 -20.27321 -13.434846
Write       5.8083167  13.95895   5.931076
```

Del mismo modo se presentan los p-valores con los valores de la distribución normal:

```
          (Intercept)  au_count2    au_count3    au_count4    n_authors
Conceive 1.261325e-01 0.23210702 1.877486e-11 1.667370e-39 1.017158e-25
Perform  3.441537e-13 0.34319383 1.207095e-19 9.347250e-83 1.648608e-26
Write    5.673164e-03 0.01771847 3.323118e-07 8.877475e-24 6.017493e-32

          n_countries n_institutes        p_age     pubs_age
Conceive 3.991489e-01 6.929170e-04 2.272079e-59 2.048434e-08
Perform  9.910280e-01 5.756835e-01 2.216962e-91 3.778542e-41
Write    1.860003e-05 6.310407e-09 2.775058e-44 3.009554e-09
```

De acuerdo con los p-valores, los parámetros no significativos y significativos en el modelo según las tareas teniendo como referencia la tarea análisis corresponderían así:

- **Tarea Concepción:**

No significativos: Intercepto, segunda posición de la firma, número de países.
Significativos: Tercera y cuarta posición de la firma, número de autores, número de instituciones, años de vida académica y número de publicaciones del autor hasta el año de publicación del artículo.

- **Tarea Desarrollo:**

No significativos: Segunda posición de la firma, número de paises, número de instituciones.
Significativos: Intercepto, Tercera y cuarta posición de la firma, número de autores, años de vida académica y número de publicaciones del autor hasta el año de publicación del artículo.

- **Tarea Escritura:**

No significativos: Segunda posición de la firma.
Significativos: Intercepto, Tercera y cuarta posición de la firma, número de países, número de instituciones, número de autores, años de vida académica y número de publicaciones del autor hasta el año de publicación del artículo.

Se identifica que la segunda posición en la firma no es significativa para ninguna tarea, sin embargo, las otras posiciones si ayudan a identificar la tarea que realiza en autor en el desarrollo del artículo.

Al igual que en la selección Stepwise donde se identificó que las variables, número de países e instituciones involucrados en el artículo eran las que menos aportaban a la reducción del AIC, se muestra que estos parámetros son los menos significativos en las tareas, por lo tanto, se podría inferir que estas dos variables no ayudan a definir la tarea realizada por el investigador y podrían salir del modelo con el fin de tener uno un poco más parsimonioso.

La significancia de los cocientes de ventajas se estudia mediante los intervalos de confianza:

Concepción 2.5 % 97.5 % (Intercept) 0.9861704 1.1195183 au_count2 0.9517535 1.0120628 au_count3 1.0920886 1.1743432 au_count4 1.2946432 1.4172010 n_authors 0.8924966 0.9250498 n_countries 0.9841852 1.0408219 n_institutes 1.0114737 1.0435505 p_age 1.0127502 1.0162764 pubs_age 1.0003958 1.0008212	**Ventaja de la tarea Concepción frente a la tarea Análisis:** Es de apreciar que los cocientes de ventaja no significativos (intervalo que contienen a 1) son: Intercepto, firma segunda posición, número de países y de manera muy cercana el número de publicaciones del autor hasta el año de publicación del artículo.
Desarrollo 2.5 % 97.5 % (Intercept) 0.7263727 0.8319303 au_count2 0.9839345 1.0476322 au_count3 0.7954508 0.8628577 au_count4 0.5366930 0.6020416 n_authors 1.0874022 1.1292637 n_countries 0.9694858 1.0311100 n_institutes 0.9782853 1.0122753 p_age 0.9737473 0.9783254 pubs_age 0.9967503 0.9975768	**Ventaja de la tarea Desarrollo frente a la tarea Análisis:** Es de apreciar que los cocientes de ventaja no significativos (intervalo que contienen a 1) son: Firma segunda posición, número de países, número de instituciones y de manera muy cercana el número de publicaciones del autor hasta el año de publicación del artículo.
Escritura 2.5 % 97.5 % (Intercept) 1.0259026 1.1617458 au_count2 0.9355062 0.9936866 au_count3 1.0591344 1.1377800 au_count4 1.2036716 1.3167428 n_authors 0.8840350 0.9157138 n_countries 1.0328199 1.0907108 n_institutes 1.0305478 1.0626233 p_age 1.0106057 1.0140948 pubs_age 1.0004287 1.0008521	**Ventaja de la tarea Escritura frente a la tarea Análisis:** Es de apreciar que no hay cocientes de ventaja no significativos (se excluye al 1 del intervalo); sin embargo, los intervalos para las variables; número de países, número de instituciones, número de publicaciones del autor hasta el año de publicación del artículo y la edad académica del autor son muy cercanos a 1 lo cual muestra su baja

	significancia en la predicción de la tarea.

A partir de la estimación de los parámetros se determinan estimaciones de probabilidad para cada categoría de la variable respuesta, a continuación se presentan las estimaciones para los primeros 10 datos de la muestra.

```
    Analysis  Conceive    Perform     Write
1  0.2507928 0.2172993 0.26413972 0.2677682
2  0.2534962 0.2128708 0.27536699 0.2582660
3  0.2625710 0.2650097 0.19368325 0.2787361
4  0.2664304 0.2185032 0.27961285 0.2354535
5  0.2631042 0.2100829 0.28837751 0.2384354
6  0.2308608 0.3362046 0.08410675 0.3488278
7  0.2670452 0.2324233 0.25218334 0.2483481
8  0.2722091 0.2486821 0.22452530 0.2545835
9  0.2668684 0.2954826 0.14824899 0.2894001
10 0.2489444 0.3137193 0.12305685 0.3142795
                                       …
```

En el análisis por fila se puede observar que las probabilidades calculadas para cada una de las cuatro tareas no presentan grandes diferencias en la mayoría de los casos, por tanto, la predicción de la categoría que corresponde a las variables explicativas que conforman las observaciones no es tan clara, es decir, la decisión de asignación de una determinada categoría se toma por un porcentaje mínimo respecto a otra de las categorías.

A continuación, se presentan las predicciones para las 10 primeras observaciones de la muestra, asignación realizada para la categoría de mayor probabilidad.

```
[1] Write    Perform    Write      Perform    Perform
[6] Write    Analysis   Analysis   Conceive   Write   …
Levels: Analysis Conceive Perform Write
```

Con el objetivo de determinar que tan bueno es el modelo en cuanto a la predicción correcta de la tarea, en cada una de las observaciones se construye la tabla de clasificación, en la cual las filas corresponden a las observaciones y las columnas a las predicciones, los resultados a continuación.

```
          Analysis Conceive Perform Write
Analysis     10581      5916   13925 22900
Conceive      9101      8222   10675 27012
Perform      10327      1922   14561 14517
Write         9670      8001   11888 28611
```

De lo anterior se concluye que el modelo hace una predicción acertada de la tarea en un 29.82% de los casos y adicionalmente se tiene que:

- Entre los autores que realizan la tarea de análisis, el modelo acierta en un 19.84%
- Entre los autores que realizan la tarea de Concepción, el modelo acierta en un 14.94%
- Entre los autores que realizan la tarea de Desarrollo, el modelo acierta en un 35.23%
- Entre los autores que realizan la tarea de Escritura, el modelo acierta en un 49.19%

4.2.2 Modelo Logit Multinomial con datos de entrenamiento

De acuerdo con los resultados obtenidos en la sección anterior, el modelo propuesto queda especificado de la siguiente forma:

$$Task = au\_count + n\_authors + p\_age + pubs\_age$$

Para la división del conjunto de datos en dos subconjuntos; Datos de entrenamiento y de prueba se sigue la regla 70-30, es decir, el 70% de los datos sirven para el entrenamiento del modelo y el 30% restante para validar el nivel de predicción del modelo.

A continuación, se presentan los coeficientes resultantes del ajuste de un modelo logístico multinomial en los datos de entrenamiento.

```
multinom(formula = Task ~ ., data = Train)
```

Coefficients:

```
         (Intercept)   au_count2   au_count3  au_count4   n_authors       p_age
Conceive   0.1143670 -0.03167933  0.11032360  0.2910794 -0.09307592  0.01432374
Perform   -0.2339597  0.01092523 -0.19602082 -0.5582149  0.09704584 -0.02385973
Write      0.2179732 -0.04631767  0.07941779  0.2214595 -0.09589298  0.01235602

             pubs_age
Conceive  0.0006533471
Perform  -0.0030662032
Write     0.0006713828
```

Std. Errors:

```
         (Intercept)  au_count2  au_count3  au_count4  n_authors       p_age
Conceive  0.03647644 0.01873846 0.02210460 0.02745053 0.01066348 0.001057968
Perform   0.03890272 0.01913504 0.02478892 0.03480830 0.01125534 0.001433528
Write     0.03583730 0.01841127 0.02182585 0.02727002 0.01048029 0.001049666
            pubs_age
Conceive 0.0001297339
Perform  0.0002573536
Write    0.0001293049

Residual Deviance: 393410.5
AIC: 393452.5
```

Al comparar la deviance residual y el valor del AIC de este nuevo modelo con el modelo que incluía todas las variables, se identifica una reducción en estas dos medidas cercana al 30%, lo cual muestra que la exclusión de estas dos variables mejora el modelo y lo deja más parsimonioso.

La interpretación de parámetros y significancia se realiza de la misma forma que en la sección anterior, por tanto, en esta sección no se presenta y se centra el interés en la predicción del modelo.

Con el objetivo de determinar que tan bueno es el modelo en cuanto a la predicción correcta de la tarea, se construye la tabla de clasificación, en la cual las filas corresponden a las observaciones y las columnas a las predicciones, inicialmente se presentan estos resultados en los datos de entrenamiento, los resultados a continuación.

```
          predtrain
           Analysis Conceive Perform Write
  Analysis     6567     2899   10189 17608
  Conceive     5667     4117    7848 20958
  Perform      6341      884   10660 11068
  Write        6121     4113    8751 21689
```

De lo anterior se concluye que el modelo hace una predicción acertada de la tarea en un 29.58% de los casos, lo cual comparado con la predicción acertada del modelo saturado no varía mucho, puesto que era del 29.82%.

Los resultados para la predicción en los datos de prueba, muestran que la predicción acertada es del 29.57% de los casos, mostrando que este porcentaje es muy similar al obtenido en los datos de entrenamiento, la matriz se presenta a continuación.

	predtest			
	Analysis	Conceive	Perform	Write
Analysis	2780	1183	4400	7696
Conceive	2416	1790	3390	8824
Perform	2783	393	4498	4700
Write	2591	1748	3788	9369

Como ya se mencionó la distribución de las probabilidades calculadas para cada una de las cuatro tareas no presentan grandes diferencias, por tanto, la predicción de la categoría que corresponde a las variables explicativas que conforman las observaciones no es tan clara, es decir, la decisión de asignación de una determinada categoría se toma por un porcentaje mínimo respecto a otra de las categorías, esto se da más por la naturaleza de los datos abordados en esta investigación.

4.3 Árboles de decisión

Los árboles de decisión son un método usado en distintas áreas como un modelo de predicción y tienen cierta similitud a los diagramas de flujo, en los cuales se puede llegar a tomar decisiones de clasificación de acuerdo a una regla.

La idea de clasificar con árboles de decisión es simple: iterativamente se irán generando particiones binarias sobre los datos de interés, en las cuales se

busca que cada nueva partición genere un subgrupo de datos, lo más homogéneo posible.

Para este caso se tiene que de acuerdo a cada una de las variables se realice una partición binaria y determine según la regla la tarea que se ejecuta por parte de los autores en la publicación de artículos científicos.

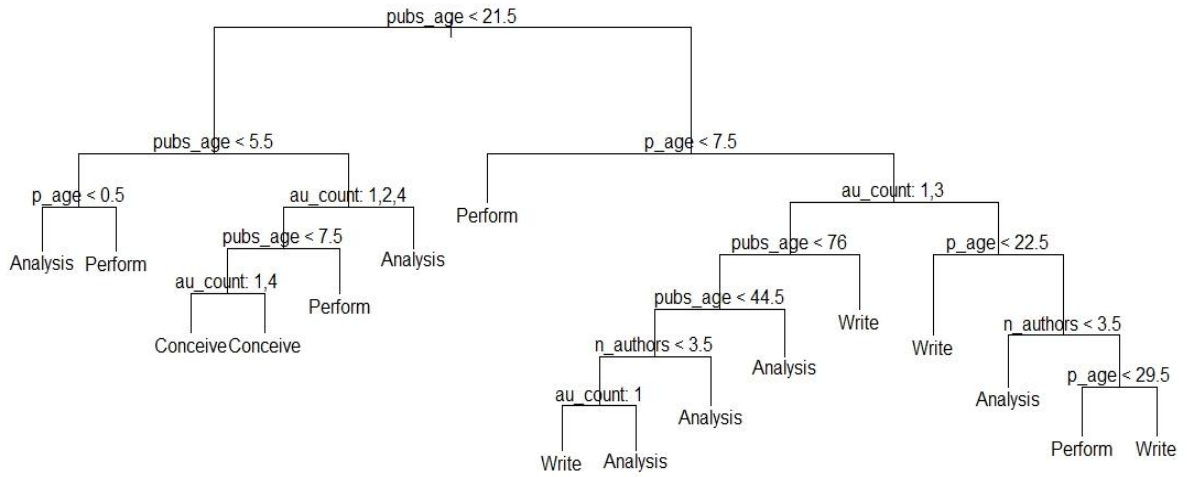
Al igual que en los modelos logísticos multinomiales inicialmente se trabaja con todas las variables y una muestra baja de entrenamiento para la construcción del árbol, posteriormente se trabaja con las variables más significativas y con un tamaño muestra de entrenamiento adecuado.

4.3.1 Árboles de decisión incluyendo todas las variables

A continuación, se presentan los resultados de la aplicación de la técnica de árboles a los datos de registros de publicación de artículos científicos. Inicialmente, se tienen en cuenta todas las variables que pueden incidir en la tarea realizada por el investigador, el árbol resultante se especifica a continuación:

Figura 13

Árboles de decisión con todas las variables



Nota: *Construido con la librería tree de R, incluyendo todas las variables*

Así mismo se presenta el resumen obtenido con la función **`summary()`**, lo cual muestra las variables que se utilizan como los nodos internos en el árbol, el número de nodos terminales y la tasa de error (entrenamiento).

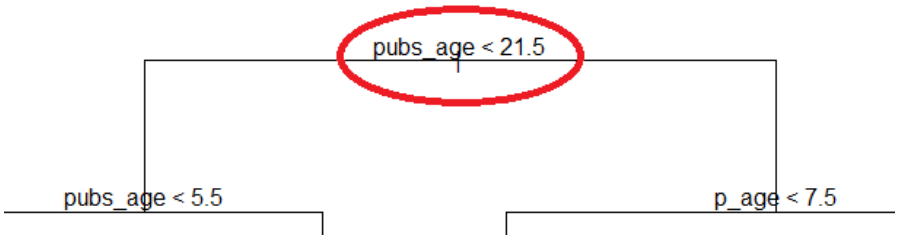
```
Classification tree:
tree(formula = Task ~ au_count + n_authors + n_countries +
n_institutes +
    p_age + pubs_age, data = Datos1, subset = train)
Variables actually used in tree construction:
[1] "pubs_age"  "p_age"     "au_count"  "n_authors"
Number of terminal nodes:  16
Residual mean deviance:  2.276 = 532.5 / 234
Misclassification error rate: 0.52 = 130 / 250
```

De lo anterior se identifica que las variables que intervienen en la construcción del árbol son: Número de publicaciones del autor hasta el año de publicación del artículo, número de años de vida académica del autor, posición de la firma y número de autores. También que el número de nodos terminales es de 16 y que se presenta una tasa de error de entrenamiento del 52%, lo ideal es que dicha tasa de error sea la menor posible, para el caso dicha tasa es menor que la de los árboles construidos inicialmente. Se utiliza este primer árbol para describir las reglas de clasificación que derivaron en la construcción de este.

Primera regla de clasificación: la variable principal que determina la primera partición de los datos corresponde a número de publicaciones del autor hasta el año de publicación del artículo y la regla indica una partición a partir de 21.5 publicaciones.

Figura 14

Árbol de decisión; Primera regla de clasificación



Nota: *Se presenta una parte del árbol, la principal regla de partición binaria*

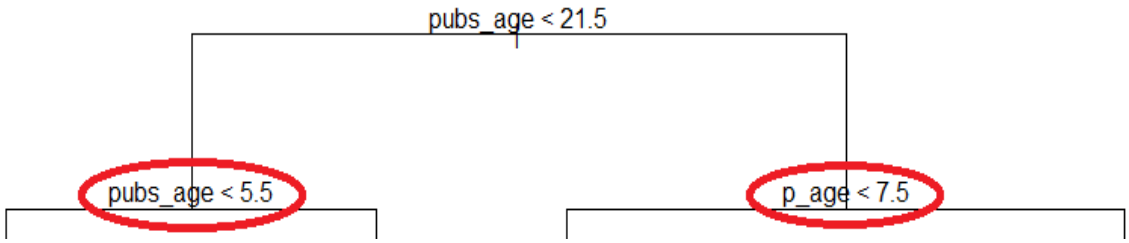
Segunda regla de clasificación: la segunda variable que permite hacer subdivisiones en uno de los grupos ya constituidos a partir de la variable número de publicaciones es la variable número de años de vida académica, mientras que en el otro subgrupo continúa la variable número de publicaciones generando nuevas subdivisiones, la regla en cada subgrupo es:

- Para número de publicaciones inferiores a 21.5 se tiene que se generan nuevos subgrupos con corte en 5.5 publicaciones en el año de publicación del artículo.

- Para número de publicaciones superiores a 21.5 se tiene que se generan subgrupos a partir de 7.5 años de vida académica del autor.

Figura 15

Árbol de decisión; Segunda regla de clasificación

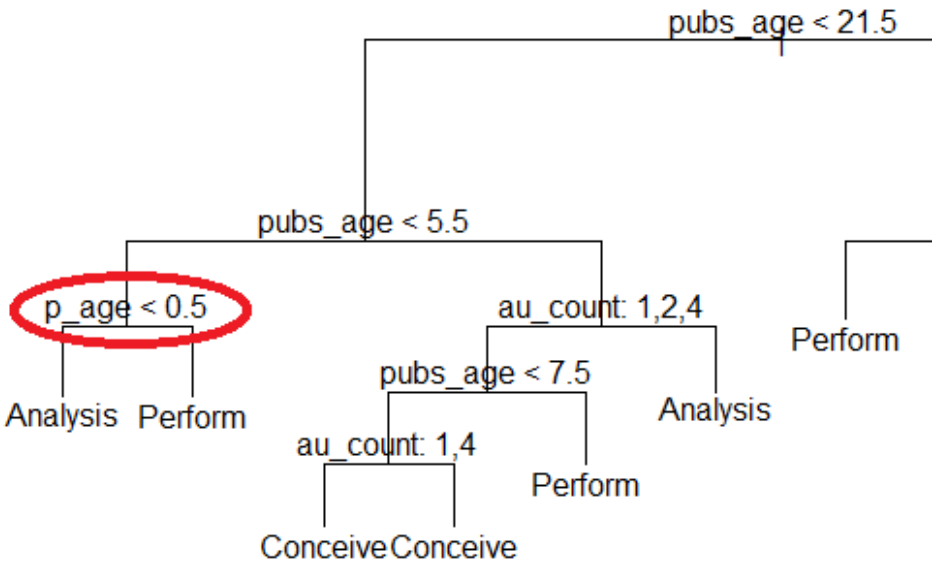


Nota: *Se presenta una parte del árbol, segunda regla de partición binaria*

De la misma manera se puede continuar el recorrido del árbol y especificar las ramas del árbol según las distintas reglas de clasificación, a modo de ejemplo se revisa la rama que está a la izquierda, es decir, la que proviene de número de publicaciones inferiores a 21.5 y luego publicaciones inferiores a 5.5.

Figura 16

Recorrido por el árbol de clasificación



Nota: *Se presenta una parte del árbol, ejemplo de otra partición binaria*

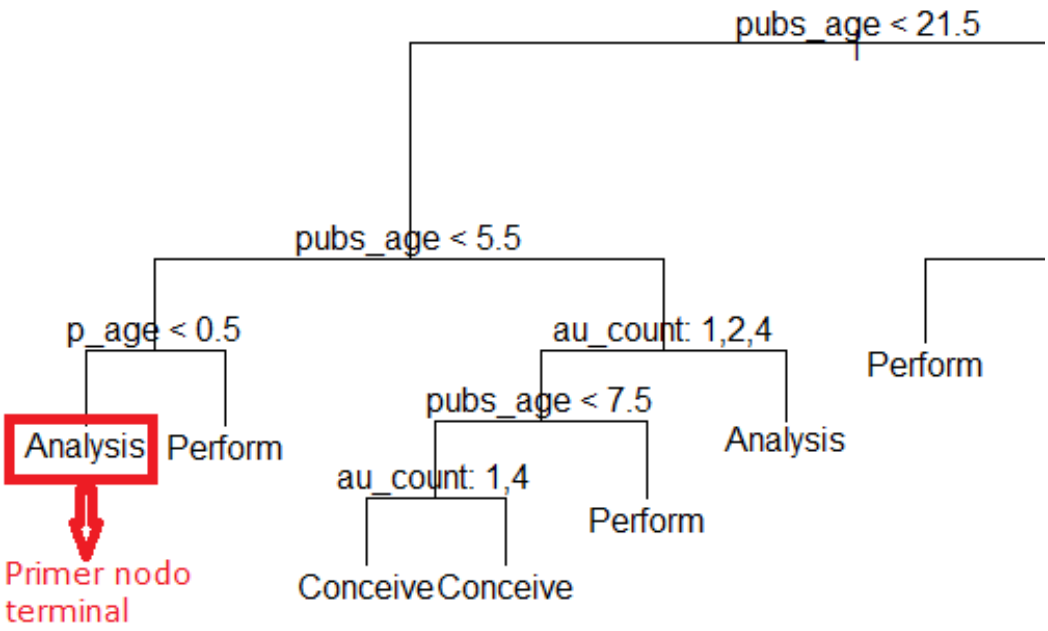
Como se observa en la gráfica anterior, los nuevos subgrupos se generan a partir de la variable número de años de vida académica del autor, con la regla de partición en 0.5 años, lo cual genera dos nodos terminales, que se especifican a continuación:

- **Primer nodo terminal:** Clasificación de la tarea desarrollada por el autor en **Análisis,** si el número de publicaciones en el año de publicación del artículo es inferior a 5.5 y número de años de vida académica del autor es inferior a 0.5 años.

- **Segundo nodo terminal:** Clasificación de la tarea desarrollada por el autor en **Desarrollo,** si el número de publicaciones en el año de publicación del artículo es inferior a 5.5 y número de años de vida académica del autor es superior a 0.5 años.

Figura 17

Nodos terminales por la izquierda del árbol

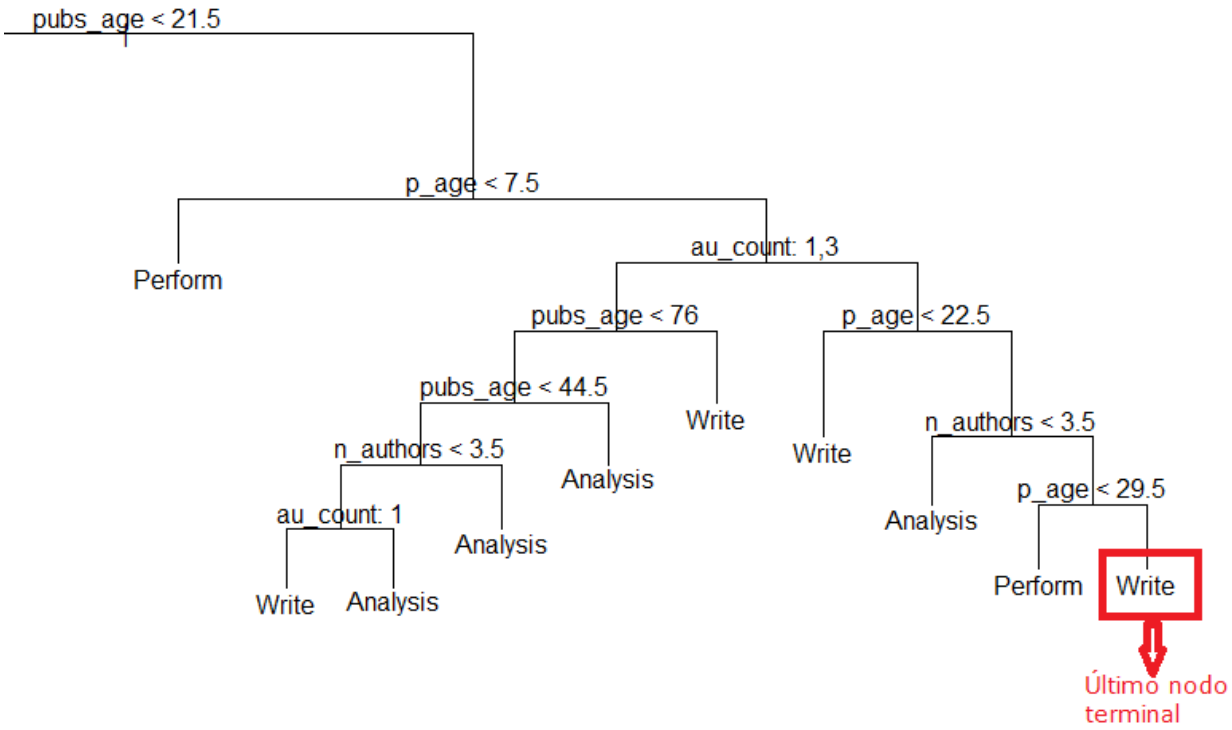


Nota: *Se presenta los dos primeros nodos terminales resultantes de la partición binaria (izquierda del árbol)*

En el contexto anterior, a continuación, se describe el último nodo terminal (de izquierda a derecha) según las reglas: Clasificación de la tarea desarrollada por el autor en **Escritura**, si el número de publicaciones en el año de publicación del artículo es superior a 21.5, el número de años de vida académica del autor es superior a 29.5 años, la posición de la firma está en los lugares 2 o 4 y el número de autores es superior a 3 (4 autores).

Figura 18

Nodo terminal final a la derecha del árbol



Nota: *Se presenta el último nodo terminal resultantes de la partición binaria (derecha del árbol)*

Dado que el conjunto de entrenamiento ha permitido la construcción del árbol especificado anteriormente, se debe determinar el porcentaje de predicciones correctas en el conjunto de prueba, obteniendo que el árbol tiene un porcentaje del 26.82% de predicciones correctas, se presenta la tabla de clasificación errónea.

```
          Task
tree.pred  Analysis Conceive Perform Write
  Analysis    11293    11212    8841 12087
  Conceive     3989     3937    3272  4183
  Perform     24916    22668   22878 24251
  Write       13061    17133    6280 17578
```

Dado que se ha construido un árbol con 16 nodos terminales y se ha presentado un porcentaje bajo de predicción correcta, se busca mejorar estos resultados mediante la poda del árbol, dicha poda se realiza utilizando validación cruzada a fin de optimizar el árbol, para el caso la función cv.tree() realiza la validación cruzada y muestra como resultados el número de nodos terminales de

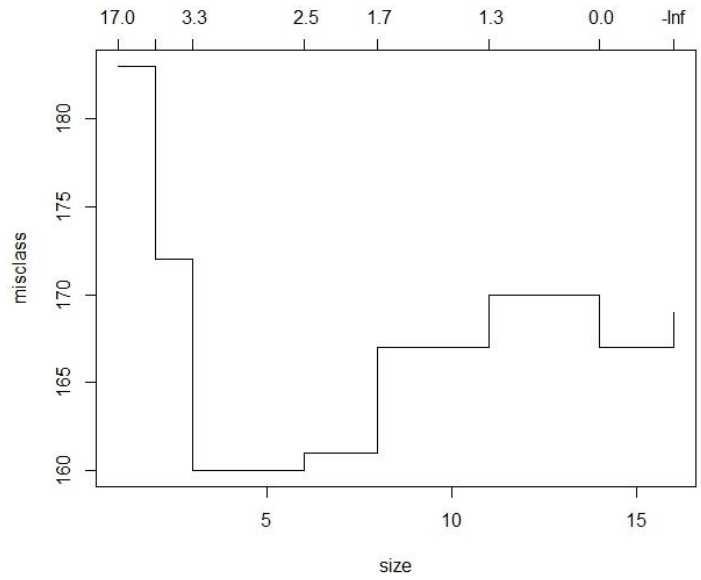
cada árbol considerado, así como la tasa de error correspondiente, los resultados son los siguientes:

```
$size
[1] 16 14 11  8  6  3  2  1

$dev
[1] 169 167 170 167 161 160 172 183

$k
[1]      -Inf  0.000000  1.333333  1.666667  2.500000  3.333333  8.000000
[8] 17.000000

$method
[1] "misclass"

attr(,"class")
[1] "prune"         "tree.sequence"
```

Esta poda muestra que el árbol con 3 nodos terminales produce la tasa de error de validación cruzada más baja, con 160 de desviación estándar en la validación cruzada (la menor). Estos resultados se pueden apreciar en la siguiente gráfica.

Figura 19

Grafica de validación cruzada

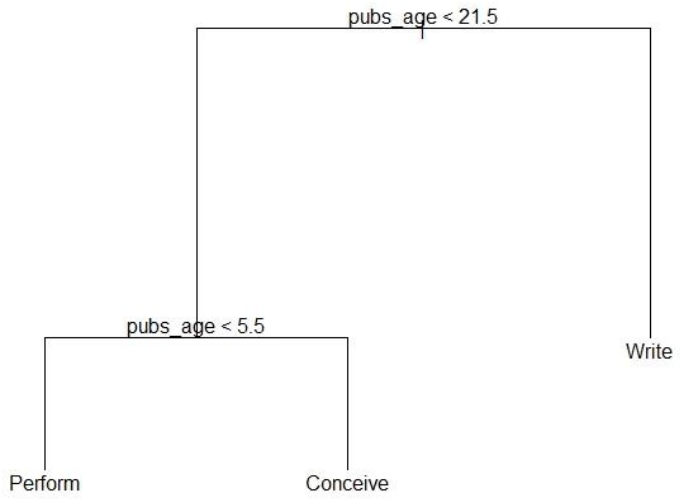


Nota: *La gráfica permite identificar el número óptimo de nodos terminales del árbol*

La función `prune.misclass()` permite realizar la poda del árbol y obtener el árbol de tres nodos terminales. Los resultados muestran un árbol más simple y de mejor interpretación que el que inicialmente se construyó con 16 nodos terminales. Adicional a esto se mejora levemente el porcentaje de predicciones correctas, puesto que dicho porcentaje pasó del 25.82% al 29.20% de predicciones correctas. A continuación, se presenta el árbol con la poda a 3 nodos terminales.

Figura 20

Árbol óptimo según validación cruzada



Nota: *Árbol óptimo con tres nodos terminales, construcción librería tree*

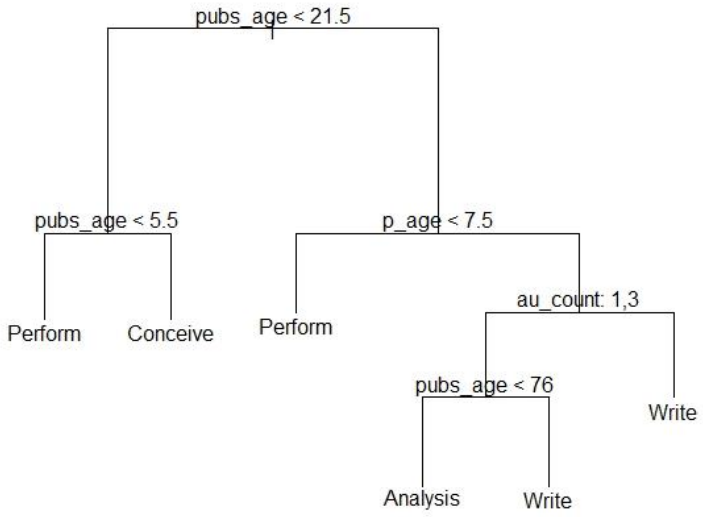
Si bien el aumento en el porcentaje de predicciones no es muy significativo, la simplificación del árbol y por consiguiente su interpretación sí lo es, dado que como podemos observar tenemos un árbol más sencillo y con un poder de predicción mayor al de los 16 nodos terminales.

Del árbol anterior se identifica que no se realiza clasificación para la tarea Análisis, debido a que en la validación cruzada el número óptimo de nodos terminales era 3 y las tareas a clasificar eran 4. Entonces la tarea Análisis queda fuera de esta clasificación. A fin de ingresar esta tarea en la clasificación del

árbol, se prueba con 4 en la validación cruzada que no está lejos del óptimo en cuanto a la tasa de error, el árbol obtenido es el siguiente.

Figura 21

Árbol con cuatro nodos terminales



Nota: *Árbol con cuatro nodos terminales, a fin de incluir las cuatro tareas analizadas*

En este árbol, si está clasificada la tarea Análisis, sin embargo, la tasa de predicción correcta cae respecto al óptimo, por tanto, no es el más adecuado.

4.3.2 Árboles de decisión con reducción de variables y datos de entrenamiento al 70%

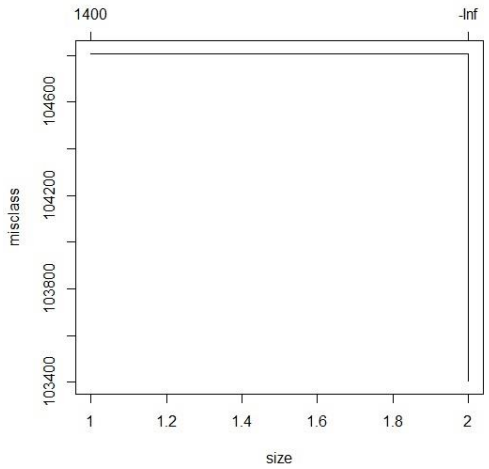
Con el fin de determinar el mejor modelo posible se recurre a la regla 70-30, en la cual el 70% de los datos son usados para el entrenamiento del árbol y el restante porcentaje se usarán para la predicción. El árbol inicial que se obtiene con los datos de entrenamiento arroja 2 nodos terminales y un porcentaje del 28.95% en la tasa de predicción correcta. Este resultado es muy cercano a los primeros árboles obtenidos con muestras de entrenamiento bajas.

Como ya se ha mencionado anteriormente, en busca de mejorar estos resultados se realiza la poda del árbol. Esta poda se realiza utilizando validación cruzada a fin de optimizar el árbol. Los resultados se presentan a continuación.

Figura 22

Validación cruzada en los árboles 70-30

```
$size
[1] 2 1

$dev
[1] 103407 104806

$k
[1] -Inf 1439

$method
[1] "misclass"

attr(,"class")
[1] "prune"         "tree.sequence"
```

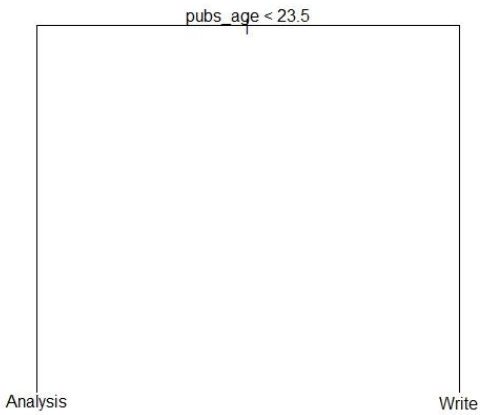


Nota: *Se busca la optimización del árbol con datos de entrenamiento al 70%*

El resultado muestra que el árbol con solo 2 nodos terminales produce la tasa de error de validación cruzada más baja. A continuación, se presenta el árbol obtenido.

Figura 23

Árbol de decisión con 70% de datos de entrenamiento



Nota: *Se construye el árbol con datos de entrenamiento al 70%, generando únicamente dos nodos terminales*

En el árbol anterior se puede inferir que la variable principal que determina la única partición de los datos corresponde a número de publicaciones hasta el año de publicación del artículo y la regla indica una partición a partir de 23.5 publicaciones. Así: si el autor tiene menos de 23.5 publicaciones hasta el año de publicación del artículo, la tarea realizada es el análisis en la investigación, por el contrario, si el autor tiene más de 23.5 publicaciones hasta el año de publicación del artículo, la tarea realizada es la de escritura del artículo científico.

Dado que el conjunto de entrenamiento ha permitido la construcción del árbol especificado anteriormente, se debe determinar el porcentaje de predicciones correctas en el conjunto de entrenamiento inicialmente, obteniendo que el árbol tiene un porcentaje del 28.95% de predicciones correctas, se presenta la tabla de clasificación errónea.

	Task			
tree.pred	Analysis	Conceive	Perform	Write
Analysis	23792	20731	22913	22353
Conceive	0	0	0	0
Perform	0	0	0	0
Write	13471	17859	6040	18321

La predicción en el conjunto de prueba muestra que el árbol hace una predicción correcta de la tarea en un 29.13% de los casos, lo que presenta un leve aumento respecto a los datos de entrenamiento, a continuación, se presenta la tabla de clasificación.

	Task			
tree.pred	Analysis	Conceive	Perform	Write
Analysis	10313	8871	9819	9646
Conceive	0	0	0	0
Perform	0	0	0	0
Write	5746	7549	2555	7850

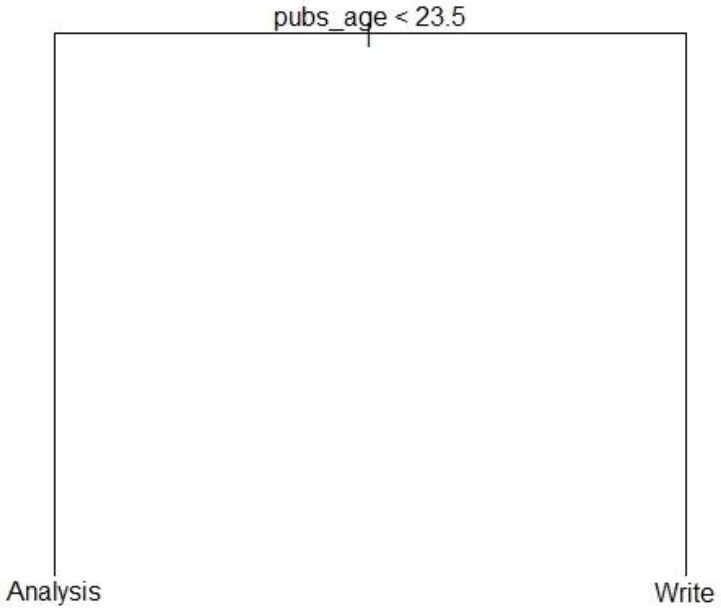
Como se puede apreciar en los resultados anteriores, el árbol resultante surge de una única partición binaria definida por una variable, en este contexto

se quiso verificar si al construir árboles en los cuales solo se incluye una única variable se generan resultados similares a lo ya obtenido.

Se identificó que las únicas variables que de manera individual logran clasificar en grupos binarios los datos son; Número de publicaciones hasta el año de publicación del artículo y número de años de vida académica del autor, la clasificación se realiza para la variable respuesta: Tarea realizada por el autor, a continuación, se muestran los resultados.

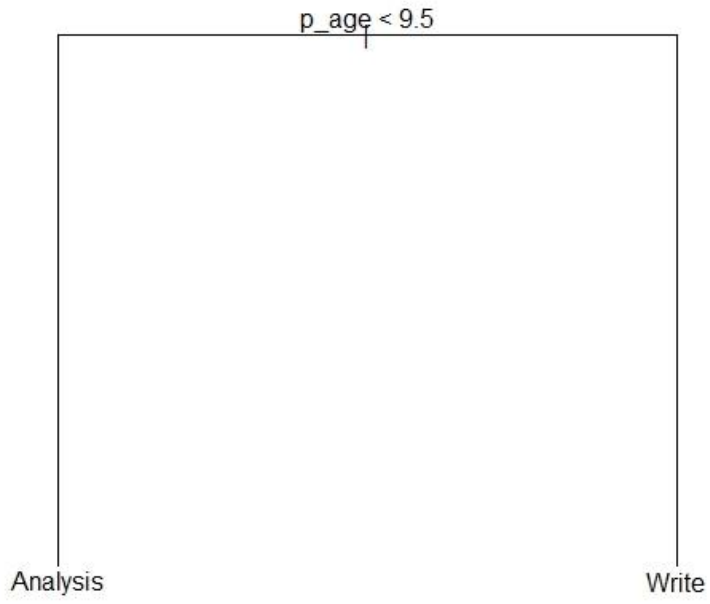
Figura 24

Árboles de decisión con una sola variable



Número de publicaciones hasta el año de publicación del artículo:

Regla de clasificación: Autores con menos de 23.5 publicaciones hasta el año de publicación del artículo realizan la tarea de Análisis, mientras que los que tienen más de 23.5 publicaciones realizan la tarea de escritura.



Número de años de vida académica del autor:

Regla de clasificación: Autores con menos de 9.5 años de vida académica realizan la tarea de Análisis, mientras que los que tienen más de 9.5 años de vida académica realizan la tarea de escritura.

Nota: *Construidos con la librería tree de R*

El resultado anterior muestra que, si solo se contempla la variable número de publicaciones hasta el año de publicación del artículo, se obtiene un resultado similar al del árbol generado cuando se tenían en cuenta más variables.

A continuación, se realiza la comparación del modelo multinomial y los árboles de decisión.

4.4 Comparación del modelo logístico multinomial y los árboles de decisión

Dado que el objetivo del trabajo estaba enfocado en la comparación de estas dos técnicas en el análisis del conjunto de datos abordados, a continuación, se presentan las tasas de clasificación errónea, tanto en los datos de entrenamiento como en los datos de prueba.

Modelo logístico multinomial	**Árbol de decisión**
Predicción acertada de la tarea en un 29.58% de los casos en los datos de entrenamiento, tabla de clasificación:	Predicción acertada de la tarea en un 28.95% de los casos en los datos de entrenamiento, tabla de clasificación:

predtrain Analysis Conceive Perform Write Analysis 6567 2899 10189 17608 Conceive 5667 4117 7848 20958 Perform 6341 884 10660 11068 Write 6121 4113 8751 21689	Task tree.pred Analysis Conceive Perform Write Analysis 23792 20731 22913 22353 Conceive 0 0 0 0 Perform 0 0 0 0 Write 13471 17859 6040 18321
Los resultados para los datos de prueba, muestran que la predicción acertada es del 29.57% de los casos, tabla de calsificación: predtest Analysis Conceive Perform Write Analysis 2780 1183 4400 7696 Conceive 2416 1790 3390 8824 Perform 2783 393 4498 4700 Write 2591 1748 3788 9369	Los resultados para los datos de prueba, muestran que la predicción acertada es del 29.13% de los casos, tabla de calsificación: Task tree.pred Analysis Conceive Perform Write Analysis 10313 8871 9819 9646 Conceive 0 0 0 0 Perform 0 0 0 0 Write 5746 7549 2555 7850

Como se puede apreciar en la comparación anterior, las dos técnicas tienen porcentajes similares en cuanto a la capacidad de predicción correcta, tanto en los datos de entrenamiento como en los datos de prueba, también se puede apreciar que el porcentaje aumenta levemente en los datos de prueba.

Sin embargo, este porcentaje es bajo con ambas técnicas, esto se da más por la naturaleza de los datos.

5. Conclusiones

Los árboles de decisión junto con otras técnicas del Data Mining se presentan como una alternativa computacional para abordar el objetivo del análisis de regresión logística, generando en algunas ocasiones representaciones gráficas sencillas de interpretar.

Las variables número de publicaciones del autor hasta el año de publicación del artículo en cuestión, número de años de vida académica del autor, número de autores y posición de la firma del autor en el artículo; son las que mayor aporte realizan a la predicción de la variable respuesta (tarea a desarrollar por el autor). Y esto tanto en los modelos logísticos multinomiales como en los árboles de decisión.

La tasa de predicción correcta en el modelo logístico es del 30.03% y en los árboles de decisión esta tasa es del 29.79% en los datos de prueba, si bien esta tasa es baja muestra la similitud entre estas dos técnicas para predecir de manera correcta la tarea a realizar por el autor en el artículo científico. Dentro de las posibles causas para tasas tan bajas se podría dar el ámbito donde se desarrollan las investigaciones, puesto que estas son desarrolladas en el campo médico y se registran abundantes publicaciones en un año para un mismo autor, esto debido a las colaboraciones que se presentan, por lo tanto, un mismo autor puede desempeñar varios roles en diferentes artículos, lo cual no permite realizar una clasificación más limpia.

En los árboles de decisión se presentan reglas de clasificación de fácil entendimiento a diferencia de la interpretación de los parámetros en el modelo logístico, sin embargo, son susceptible a ser inestables, puesto que, un cambio en los datos (tamaño de la muestra o semilla) pueden modificar la estructura del árbol, lo que hace que la interpretación de las reglas cambie de un árbol a otro, así que como una posible línea de trabajo para hacer robustas las decisiones

tomadas a partir de un árbol se propone agregar muchos árboles de decisión y combinar sus resultados por los métodos de Embolsado, bosques aleatorios, entre otros métodos.

Finalmente, también se propone como futuros trabajos el análisis de datos categóricos por medio de otras técnicas del Data Mining y su comparación con las tasas de predicción correcta del modelo logístico.

6. Bibliografía

Agresti, A. (1996). An introduction to categorical data analysis. Jhon Wiley, New York

Agresti, A. (2002). Categorical Data Analysis (2ª edition). Wiley.

Aguilera, A. M. y Escabias, M. (2021), Modelos de respuesta discreta. Aplicaciones Biosanitarias.

Box, G. E. y Tidwell, P. W. (1962), Transformation of the independent variables. Technometrics 4 (4). 531-550

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984), Classification and Regression Trees, Wadsworth. Belmont.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. Classification and Regression Trees. 1nd ed. Boca Raton, Florida: Chapman; Hall/CRC. https://www.crcpress.com/Classification-and-Regression-Trees/Breiman-Friedman-Stone-Olshen/p/book/9780412048418.

Choque, G. (2009). Árboles de decisión. Mente Errabunda. http://menteerrabunda.blogspot.com/2009/05/arboles-de-decision.

Díaz, L. G. Morales, M. A. (2009). Análisis de datos categóricos (Primera ed.). Editorial Universidad Nacional de Colombia.

Hernández, F. (2021). Modelos Predictivos. https://fhernanb.github.io/libro_mod_pred/

Gareth J., Witten D., Hastie T., and Tibshirani R. (2013). An Introduction to Statistical Learning – with Applications in r. Vol. 103. Springer Texts in Statistics. New York, Springer.

McCullagh, P. and Nelder, J. A. (1989). Generalized linear models. 2ed. Chapman and Hall, New York.

Nelder, J. A. and Wedderburn, R. W. (1972). Generalized linear models. Journal of the Royal Statistical Society: Series A. 135 (3). 370-384.

R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Ripley, B (2019). Tree: Classification and Regression Trees. https://CRAN.R-project.org/package=tree.

Robinson-Garcia, N., Costas, R., Sugimoto, C., Larivière, V., Nane G. (2020) Meta-Research: Task specialization across research careers eLife 9:e60586

Robinson-Garcia, N., Costas, R., Sugimoto, C., Larivière, V., Nane G. (2020) Datasets used in the study 'Task specialization and its effects on research careers'. https://zenodo.org/record/3891055#.Y964LNLMLlg

Therneau, T., and Atkinson B (2019). Rpart: Recursive Partitioning and Regression Trees. https://CRAN.R-project.org/package=rpart.

Venables, W. N. y Ripley, B. D. (2002) Modern Applied Statistics with S. Cuarta edición. Salmer.

Wu, X., Kumar, V., Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., (…) Steinberg, D. (2008). Top 10 algorithms in data mining. Knowledge Informatics Systems, 14, 1-37. DOI: 10.1007/s10115-007-0114-2

Anexo A

CÓDIGO EN R

```
#Importar datos
Datos<-read.csv(file="PlosTFMF.csv",header = TRUE,sep=",")

#Sin investigador y artículo
Task<-as.factor(Datos$Task)
au_count<-as.factor(Datos$au_count)
n_authors<-Datos$n_authors
n_countries<-Datos$n_countries
n_institutes<-Datos$n_institutes
p_age<-Datos$p_age
pubs_age<-Datos$pubs_age

Datos1<-
data.frame(au_count,n_authors,n_countries,n_institutes,p_age,
       pubs_age,Task)

#Análisis descriptivo
table(Task)
barplot(((table(Task)/length(Task))*100),col=rgb(0.7, 0.87, 0.93),
        main="Tarea realizada por el investigador",ylab="%")

(table(au_count)/length(au_count))*100
barplot(((table(au_count)/length(au_count))*100),col=rgb(0.7, 0.87,
0.93),
        main="Posición de la firma del investigador",ylab="%")

(table(n_authors)/length(n_authors))*100
summary(n_authors)
sd(n_authors)
cv=sd(n_authors)/mean(n_authors)
barplot(((table(n_authors)/length(n_authors))*100),col=rgb(0.7,
0.87, 0.93),
        main="Número de autores en la publicación",ylab="%")

(table(n_countries)/length(n_countries))*100
barplot(((table(n_countries)/length(n_countries))*100),col=rgb(0.7,
0.87, 0.93),
        main="Número de países involucrados en la
investigación",ylab="%",ylim=c(0,80))
summary(n_countries)
sd(n_countries)
cv=sd(n_countries)/mean(n_countries)

(table(n_institutes)/length(n_institutes))*100
barplot(((table(n_institutes)/length(n_institutes))*100),col=rgb(0.7
, 0.87, 0.93),
        main="Número de institutos involucrados en la
investigación",ylab="%",ylim=c(0,50))

(table(p_age)/length(p_age))*100
summary(p_age)
boxplot(p_age,col=rgb(0.7, 0.87, 0.93),main="Edad académica",
```

```
        outpch = 20,outbg = "blue")
hist(p_age,col=rgb(0.7, 0.87, 0.93),main="Edad académica",
     outpch = 20,outbg = "blue")

round((table(pubs_age)/length(pubs_age))*100,2)
summary(pubs_age)
sd(pubs_age)
cv=sd(pubs_age)/mean(pubs_age)
boxplot(pubs_age,col=rgb(0.7, 0.87, 0.93),main="Número de
publicaciones en el año",
        outpch = 20,outbg = "blue")
hist(pubs_age,col=rgb(0.7, 0.87, 0.93),main="Número de publicaciones
en el año",
     outpch = 20,outbg = "blue",ylim=c(0,200000))

## Matriz de correlación
Datos2<-
data.frame(n_authors,n_countries,n_institutes,p_age,pubs_age)
corrplot.mixed(cor(Datos2),lower = "number",upper = "circle",tl.col
= "black")

######################
##Modelo Logístico####

####Sin datos de entrenamiento

library(nnet)
ModTar<-multinom(Task~.,data=Datos1)
summary(ModTar)
exp(summary(ModTar)$coefficients)

# Selección Stepwise
ModTar.0<-multinom(Task~1,data=Datos1)
ModTar.Step<-step(ModTar.0,
scope=list(lower=Task~1,upper=Task~au_count + n_authors +
n_countries +
           n_institutes + p_age + pubs_age,Task), direction="both")

#valores experimentales del test de Wald
summary(ModTar)$coefficients/
summary(ModTar)$standard.errors

# p-valores con las probabilidades de la distribución normal
2*pnorm(abs(summary(ModTar)$coefficients/
summary(ModTar)$standard.errors),lower.tail=F)

# Significación de los cocientes de ventajas,
# intervalos de confianza:
exp(confint(ModTar))

#Probabilidades predichas, para el 30% de los datos
predict(ModTar,type="probs")[1:62349,]

#el modelo predice, para cada valor/es de la variable/s
explicativa/s
#la categoría de la respuesta con mayor probabilidad
```

```
predict(ModTar,type="class")[1:62349]

#tabla de clasificación
table(Datos1$Task, predict(ModTar,type="class"))

#Bondad de ajuste
Aj.Mult.Cuanti$deviance
#p-valor de la chi
pchisq(Aj.Mult.Cuanti$deviance,591,lower.tail = F)

###Con datos de entrenamiento###

Datos2<-data.frame(au_count,n_authors,p_age,pubs_age,Task)
#Datos de entrenamiento y prueba

set.seed(10)
train<-sample(1:nrow(Datos2),0.7*nrow(Datos2))
dim(Datos2[train,])
dim(Datos2[-train,])

Train<-Datos2[train,]
Test<-Datos2[-train,]

################################

ModTar2<-multinom(Task~.,data=Train)
summary(ModTar2)
exp(summary(ModTar2)$coefficients)

##predicción en datos de entrenamiento
predtrain<-predict(ModTar2, newdata = Train, "class")
tablapred <- table(Train$Task, predtrain)
round((sum(diag(tablapred))/sum(tablapred))*100,2)

##predicción en los datos de prueba
predtest<-predict(ModTar2, newdata = Test, "class")
tablatest<- table(Test$Task,predtest)
round((sum(diag(tablatest))/sum(tablatest))*100,2)

#########################
## ÁRBOLES DE DECISIÓN ##
library(ISLR)
library(tree)

#Árbol
tree.Task = tree(Task ~ au_count + n_authors + n_countries +
n_institutes +
                 p_age + pubs_age, data=Datos1)
summary(tree.Task)
plot(tree.Task)
text(tree.Task, pretty = 0)
tree.Task
##################árboles por variable######
tree.Task = tree(Task ~ pubs_age, data=Datos1)
tree.Task = tree(Task ~ au_count, data=Datos1)
tree.Task = tree(Task ~ n_authors, data=Datos1)
```

```
tree.Task = tree(Task ~ n_countries, data=Datos1)
tree.Task = tree(Task ~ n_institutes, data=Datos1)
tree.Task = tree(Task ~ p_age, data=Datos1)
#################
summary(tree.Task)
plot(tree.Task)
text(tree.Task, pretty = 0)
tree.Task

####Muestra de entrenamiento####
set.seed(101)
train=sample(1:nrow(Datos1), 250)
tree.Task1 = tree(Task ~ au_count + n_authors + n_countries +
n_institutes +
                  p_age + pubs_age, data=Datos1, subset=train)
plot(tree.Task1)
text(tree.Task1, pretty = 0)

#Predicción en conjunto de prueba
tree.pred = predict(tree.Task1, Datos1[-train,], type="class")

#Tabla de clasificación erronea
with(Datos1[-train,], table(tree.pred, Task))

#Poda con clasificación erronea
cv.Datos1 = cv.tree(tree.Task1, FUN = prune.misclass)
cv.Datos1
plot(cv.Datos1)

#Árbol con 3 pasos hacia abajo
prune.Datos1 = prune.misclass(tree.Task1, best = 3)
plot(prune.Datos1)
text(prune.Datos1, pretty=0)

#En el conjunto de prueba
tree.pred = predict(prune.Datos1, Datos1[-train,], type="class")
with(Datos1[-train,], table(tree.pred, Task))

#################
#árboles regla 70:30

Datos2<-data.frame(au_count,n_authors,p_age,pubs_age,Task)

tree.Task2 = tree(Task ~ au_count + n_authors + p_age + pubs_age,
                  data=Train)
plot(tree.Task2)
text(tree.Task2, pretty = 0)

#Predicción en conjunto de prueba
tree.pred = predict(tree.Task2, newdata=Train, type="class")

#Tabla de clasificación erronea
treepred<-with(Train, table(tree.pred, Task))
round((sum(diag(treepred))/sum(treepred))*100,2)

#Poda con clasificación erronea
```

```
cv.Datos2 = cv.tree(tree.Task2, FUN = prune.misclass)
cv.Datos2
plot(cv.Datos2)

#Árbol con 2 pasos hacia abajo
prune.Datos2 = prune.misclass(tree.Task2, best = 2)
plot(prune.Datos2)
text(prune.Datos2, pretty=0)

#En el conjunto de entrenamiento
tree.pred = predict(tree.Task2, newdata=Train, type="class")
treepred<-with(Train, table(tree.pred, Task))
round((sum(diag(treepred))/sum(treepred))*100,2)

#En el conjunto de prueba
tree.pred = predict(tree.Task2, newdata=Test, type="class")
treepred<-with(Test, table(tree.pred, Task))
round((sum(diag(treepred))/sum(treepred))*100,2)
```

Anexo B:

Selección Stepwise

```
Start:  AIC=572963.4
Task ~ 1

trying + au_count
# weights:  20 (12
variable)
initial  value
288112.170777
iter  10 value
285041.787110
final  value 284181.733072
converged
trying + n_authors
# weights:  12 (6
variable)
initial  value
288112.170777
iter  10 value
286417.101730
iter  10 value
286417.101130
iter  10 value
286417.101129
final  value 286417.101129
converged
trying + n_countries
# weights:  12 (6
variable)
initial  value
288112.170777
iter  10 value
286403.415664
iter  10 value
286403.413019
iter  10 value
286403.413019
final  value 286403.413019
converged
trying + n_institutes
# weights:  12 (6
variable)
initial  value
288112.170777
iter  10 value
286396.598672
iter  10 value
286396.598655
iter  10 value
286396.598655
final  value 286396.598655
converged
trying + p_age
# weights:  12 (6
variable)
initial  value
288112.170777
iter  10 value
282011.314956
iter  10 value
282011.313514
iter  10 value
282011.313514
final  value 282011.313514
converged
```

```
Step:  AIC=564034.6
Task ~ p_age

trying - p_age
# weights:  8 (3 variable)
initial  value
288112.170777
final  value 286478.678678
converged
trying + au_count
# weights:  24 (15
variable)
initial  value
288112.170777
iter  10 value
282610.261248
iter  20 value
281482.446834
final  value 281482.442487
converged
trying + n_authors
# weights:  16 (9
variable)
initial  value
288112.170777
iter  10 value
282308.252932
final  value 281959.509394
converged
trying + n_countries
# weights:  16 (9
variable)
initial  value
288112.170777
iter  10 value
282303.290298
final  value 281969.054708
converged
trying + n_institutes
# weights:  16 (9
variable)
initial  value
288112.170777
iter  10 value
282342.626836
final  value 281968.901253
converged
trying + pubs_age
# weights:  16 (9
variable)
initial  value
288112.170777
iter  10 value
282092.710363
final  value 281765.495968
converged
                Df
AIC
+ +au_count     15
562994.9
+ +pubs_age      9
563549.0
+ +n_authors     9
563937.0
```

```
Step:  AIC=562994.9
Task ~ p_age + au_count

trying - p_age
# weights:  20 (12
variable)
initial  value
288112.170777
iter  10 value
285041.787110
final  value 284181.733072
converged
trying - au_count
# weights:  12 (6
variable)
initial  value
288112.170777
iter  10 value
282011.314956
iter  10 value
282011.313514
iter  10 value
282011.313514
final  value 282011.313514
converged
trying + n_authors
# weights:  28 (18
variable)
initial  value
288112.170777
iter  10 value
281724.163846
iter  20 value
281381.862639
final  value 281219.436318
converged
trying + n_countries
# weights:  28 (18
variable)
initial  value
288112.170777
iter  10 value
281910.556327
iter  20 value
281519.412405
final  value 281444.204161
converged
trying + n_institutes
# weights:  28 (18
variable)
initial  value
288112.170777
iter  10 value
281925.388078
iter  20 value
281535.711399
final  value 281448.591429
converged
trying + pubs_age
# weights:  28 (18
variable)
initial  value
288112.170777
iter  10 value
```

```
trying + pubs_age
# weights:  12 (6
variable)
initial  value
288112.170777
iter  10 value
282822.112676
final  value 282822.088888
converged
               Df
AIC
+ +p_age        6
564034.6
+ +pubs_age     6
565656.2
+ +au_count    12
568387.5
+ +n_institutes  6
572805.2
+ +n_countries   6
572818.8
+ +n_authors     6
572846.2
<none>          3
572963.4
# weights:  12 (6
variable)
initial  value
288112.170777
iter  10 value
282011.314956
iter  10 value
282011.313514
iter  10 value
282011.313514
final  value 282011.313514
converged
```

```
+ +n_institutes  9
563955.8
+ +n_countries   9
563956.1
<none>          6
564034.6
- p_age         3
572963.4
# weights:  24 (15
variable)
initial  value
288112.170777
iter  10 value
282610.261248
iter  20 value
281482.446834
final  value 281482.442487
converged
```

```
281823.876340
iter  20 value
281349.631501
final  value 281286.102157
converged
               Df
AIC
+ +n_authors    18
562474.9
+ +pubs_age     18
562608.2
+ +n_countries  18
562924.4
+ +n_institutes 18
562933.2
<none>          15
562994.9
- au_count       6
564034.6
- p_age         12
568387.5
# weights:  28 (18
variable)
initial  value
288112.170777
iter  10 value
281724.163846
iter  20 value
281381.862639
final  value 281219.436318
converged
```

```
Step:  AIC=562474.9
Task ~ p_age + au_count +
n_authors

trying - p_age
# weights:  24 (15
variable)
initial  value
288112.170777
iter  10 value
284147.963475
iter  20 value
283552.765378
iter  20 value
283552.765324
iter  20 value
283552.765323
final  value 283552.765323
converged
trying - au_count
# weights:  16 (9
variable)
initial  value
288112.170777
iter  10 value
282308.252932
final  value 281959.509394
converged
```

```
Step:  AIC=562086.8
Task ~ p_age + au_count +
n_authors + pubs_age

trying - p_age
# weights:  28 (18
variable)
initial  value
288112.170777
iter  10 value
281956.129896
iter  20 value
281779.059196
final  value 281678.499383
converged
trying - au_count
# weights:  20 (12
variable)
initial  value
288112.170777
iter  10 value
282018.338910
final  value 281705.714964
converged
trying - n_authors
# weights:  28 (18
variable)
initial  value
```

```
Step:  AIC=561954.7
Task ~ p_age + au_count +
n_authors + pubs_age +
n_institutes

trying - p_age
# weights:  32 (21
variable)
initial  value
288112.170777
iter  10 value
282036.309640
iter  20 value
281770.893175
final  value 281599.215039
converged
trying - au_count
# weights:  24 (15
variable)
initial  value
288112.170777
iter  10 value
281958.479022
iter  20 value
281645.746844
final  value 281645.738618
converged
trying - n_authors
```

```
trying - n_authors
# weights:  24 (15
variable)
initial  value
288112.170777
iter  10 value
282610.261248
iter  20 value
281482.446834
final  value 281482.442487
converged
trying + n_countries
# weights:  32 (21
variable)
initial  value
288112.170777
iter  10 value
281779.294386
iter  20 value
281523.513996
final  value 281159.813255
converged
trying + n_institutes
# weights:  32 (21
variable)
initial  value
288112.170777
iter  10 value
281661.118056
iter  20 value
281443.781580
final  value 281144.930534
converged
trying + pubs_age
# weights:  32 (21
variable)
initial  value
288112.170777
iter  10 value
281887.193882
iter  20 value
281367.402738
final  value 281022.406518
converged
                Df
AIC
+ +pubs_age     21
562086.8
+ +n_institutes 21
562331.9
+ +n_countries  21
562361.6
<none>          18
562474.9
- n_authors     15
562994.9
- au_count       9
563937.0
- p_age         15
567135.5
# weights:  32 (21
variable)
initial  value
288112.170777
iter  10 value
281887.193882
iter  20 value
281367.402738
```

```
288112.170777
iter  10 value
281823.876340
iter  20 value
281349.631501
final  value 281286.102157
converged
trying - pubs_age
# weights:  28 (18
variable)
initial  value
288112.170777
iter  10 value
281724.163846
iter  20 value
281381.862639
final  value 281219.436318
converged
trying + n_countries
# weights:  36 (24
variable)
initial  value
288112.170777
iter  10 value
281842.162297
iter  20 value
281412.750592
iter  30 value
280965.746615
iter  30 value
280965.746451
iter  30 value
280965.746451
final  value 280965.746451
converged
trying + n_institutes
# weights:  36 (24
variable)
initial  value
288112.170777
iter  10 value
281809.476080
iter  20 value
281290.100925
iter  30 value
280953.357667
iter  30 value
280953.357571
iter  30 value
280953.357571
final  value 280953.357571
converged
                Df
AIC
+ +n_institutes 24
561954.7
+ +n_countries  24
561979.5
<none>          21
562086.8
- pubs_age      18
562474.9
- n_authors     18
562608.2
- p_age         18
563393.0
- au_count      12
563435.4
```

```
# weights:  32 (21
variable)
initial  value
288112.170777
iter  10 value
281726.283281
iter  20 value
281580.224009
final  value 281254.564394
converged
trying - pubs_age
# weights:  32 (21
variable)
initial  value
288112.170777
iter  10 value
281661.118056
iter  20 value
281443.781580
final  value 281144.930534
converged
trying - n_institutes
# weights:  32 (21
variable)
initial  value
288112.170777
iter  10 value
281887.193882
iter  20 value
281367.402738
final  value 281022.406518
converged
trying + n_countries
# weights:  40 (27
variable)
initial  value
288112.170777
iter  10 value
281776.855640
iter  20 value
281258.750621
iter  30 value
280950.878787
final  value 280940.963235
converged
                Df      AIC
+ +n_countries 27 561935.9
<none>         24 561954.7
- n_institutes 21 562086.8
- pubs_age     21 562331.9
- n_authors    21 562551.1
- p_age        21 563240.4
- au_count     15 563321.5
# weights:  40 (27
variable)
initial  value
288112.170777
iter  10 value
281776.855640
iter  20 value
281258.750621
final  value 280940.963235
```

final value 281022.406518 converged	# weights: 36 (24 variable)	

```
Step:  AIC=561935.9
Task ~ p_age + au_count + n_authors +
pubs_age + n_institutes + n_countries

trying - p_age
# weights:  36 (24 variable)
initial  value 288112.170777
iter  10 value 282027.589765
iter  20 value 281792.741877
final  value 281585.239873
converged
trying - au_count
# weights:  28 (18 variable)
initial  value 288112.170777
iter  10 value 281933.128521
iter  20 value 281663.651651
final  value 281634.234333
converged
trying - n_authors
# weights:  36 (24 variable)
initial  value 288112.170777
iter  10 value 281794.871707
iter  20 value 281606.195864
iter  30 value 281242.521012
iter  30 value 281242.520975
final  value 281242.520975
converged
trying - pubs_age
# weights:  36 (24 variable)
initial  value 288112.170777
iter  10 value 281621.530347
iter  20 value 281482.026368
final  value 281132.364777
converged
```

```
trying - n_institutes
# weights:  36 (24 variable)
initial  value 288112.170777
iter  10 value 281842.162297
iter  20 value 281412.750592
iter  30 value 280965.746615
iter  30 value 280965.746451
final  value 280965.746451
converged
trying - n_countries
# weights:  36 (24 variable)
initial  value 288112.170777
iter  10 value 281809.476080
iter  20 value 281290.100925
iter  30 value 280953.357667
iter  30 value 280953.357571
final  value 280953.357571
converged
               Df      AIC
<none>         27 561935.9
- n_countries  24 561954.7
- n_institutes 24 561979.5
- pubs_age     24 562312.7
- n_authors    24 562533.0
- p_age        24 563218.5
- au_count     18 563304.5
```