ESTUDIOS DEL TRABAJO Nº 61
Enero-Junio 2021
Recibido: 20/10/2020/Aceptado: 19/5/2021

Métodos de *Machine Learning* como alternativa para la imputación de datos perdidos. Un ejercicio en base a la Encuesta Permanente de Hogares

Germán Rosati

Resumen
El presente trabajo expone algunos avances en la construcción de un modelo de imputación de valores perdidos y sin respuesta para las variables de ingreso en encuestas a hogares. Se presentan los resultados de algunos experimentos de imputación de los ingresos correspondientes a la ocupación principal de la Encuesta Permanente de Hogares, basados en técnicas de *Ensamble Learning* y *Deep Learning*: *Random Forest*, *XGBoost* y *Multi-Layer Perceptron*. Se compara la performance de estas técnicas con el método *Hot Deck* (uno de los métodos usados por el Sistema Estadístico Nacional).
En la primera y segunda parte del documento se plantea el problema de forma más específica y se pasa revista a los principales mecanismos de generación de los valores perdidos y sus consecuencias al momento de la imputación de valores perdidos. En la tercera parte, se presentan las técnicas propuestas y sus fundamentos teóricos-metodológicos. Finalmente, en la cuarta sección, se presentan los principales resultados de la aplicación de los métodos propuestos sobre datos de la Encuesta Permanente de Hogares.

Palabras clave: aprendizaje automático, datos perdidos, imputación, encuestas.

Machine Learning as alternative methods for missing data imputation. An exercise using Permanent Household Survey

Germán Rosati

Abstract
This paper presents some advances in the construction of a model for the imputation of missing values and no response for the income variables in household surveys. The results of some imputation experiments of the labor income variable of the Permanent Household Survey are presented, based on Assembly Learning and Deep Learning techniques: Random Forest, XGBoost and Multi-Layer Perceptron. The performance of these techniques is compared with the Hot Deck method (one of the methods used by the National Statistical System). In the first and second part of the document, it raises the problem more specifically and reviews the main mechanisms for generating lost values and their consequences at the time of imputation of lost values. In the third part, the proposed techniques and their theoretical-methodological foundations are presented. Finally, in the fourth section, the main results of the application of the proposed methods on data from the Permanent Household Survey are presented.

Keywords: Machine Learning, missing data, imputation, survey.

ESTUDIOS DEL TRABAJO Nº 61
Enero-Junio 2021
Recibido: 20/10/2020/Aceptado: 19/5/2021

Métodos de *Machine Learning* como alternativa para la imputación de datos perdidos. Un ejercicio en base a la Encuesta Permanente de Hogares[1]

Germán Rosati[2]

Introducción

La presencia de no respuesta y valores perdidos representa un problema recurrente en el análisis estadístico; afecta tanto a las estadísticas oficiales (encuestas a hogares, datos censales, etc.) como a los registros administrativos (de empresas u organismos) y, en términos generales, a cualquier conjunto de datos sobre el cual se busque realizar algún análisis cuantitativo. A su vez, las nuevas fuentes de datos englobadas bajo el término impreciso de *big data* –vinculadas a la utilización de tecnologías *mobile*, *logs* de navegación de sitios web, *scraping* de sitios, redes sociales, etc.– se caracterizan por formatos y procesos de producción no estructurados, orígenes diversos e inconsistencias varias. La expansión en el uso de este tipo de fuentes (Salganik, 2018) muestra la necesidad de contar con herramientas que permitan lidiar con la existencia de datos perdidos de manera performante. Este punto tiene particular relevancia dado que las rutinas de los paquetes estadísticos habitualmente utilizados en ciencias sociales asumen que se trabaja con datos completos y los métodos de imputación que utilizan pueden ser inapropiados (Medina y Galván, 2007).

El presente trabajo expone algunos avances iniciales en la evaluación de diferentes modelos para la imputación de valores perdidos en variables de ingreso de encuestas a hogares. Continúa una línea de trabajo enmarcada en un proyecto más general[3] que busca evaluar la

[1] Proyecto financiado por un subsidio (PI 2018-2019) de la Universidad Nacional de Tres de Febrero. El equipo de trabajo se encuentra conformado por Germán Rosati (director), Hugo Delfino (codirector), María Giselle Galli, Adriana Chazarreta, Anabella Caputti y Julia Gentile (investigadores). Una versión preliminar de este trabajo fue presentada en el Congreso de Estudios del Trabajo de 2019. En esta versión se actualizan algunos resultados luego de nuevas iteraciones en los modelos.

[2] IDAES-UNSAM/CONICET/UNTREF/PIMSA. E-mail de contacto: german.rosati@gmail.com.

[3] Un primer ejercicio usando un modelo basado en un ensamble de regresiones regularizadas vía LASSO y un desarrollo de la metodología de trabajo aplicada aquí puede verse en Rosati (2017). El proyecto recibe el apoyo

Recibido: 20/10/2020/Aceptado: 19/5/2021

capacidad de algunas técnicas de *Machine Learning* para la imputación de datos perdidos en variables cuantitativas y cualitativas. Se presentan los resultados preliminares de algunos experimentos de imputación de la variable de ingresos de la ocupación principal de la Encuesta Permanente de Hogares, basados en técnicas de *Ensamble Learning* y *Deep Learning*: *Random Forest*, *XGBoost* y *Multi-Layer Perceptron*. Se compara la performance de esas técnicas contrastándolas con el método llamado *Hot Deck* (uno de los métodos usados por el Sistema Estadístico Nacional).

Los métodos explorados resultan, en principio, aplicables para la imputación de cualquier tipo de valores perdidos (en variables cualitativas o cuantitativas) y para diversas fuentes de datos. Sin embargo, dada la relevancia particular que presenta el problema de la no respuesta de ingresos en encuestas a hogares (tanto en la Argentina como a nivel internacional) se presenta aquí una aplicación al respecto utilizando microdatos correspondientes al segundo trimestre de 2015[4] de la Encuesta Permanente de Hogares (EPH), relevamiento elaborado por el Instituto Nacional de Estadísticas y Censos de la Argentina (INDEC).

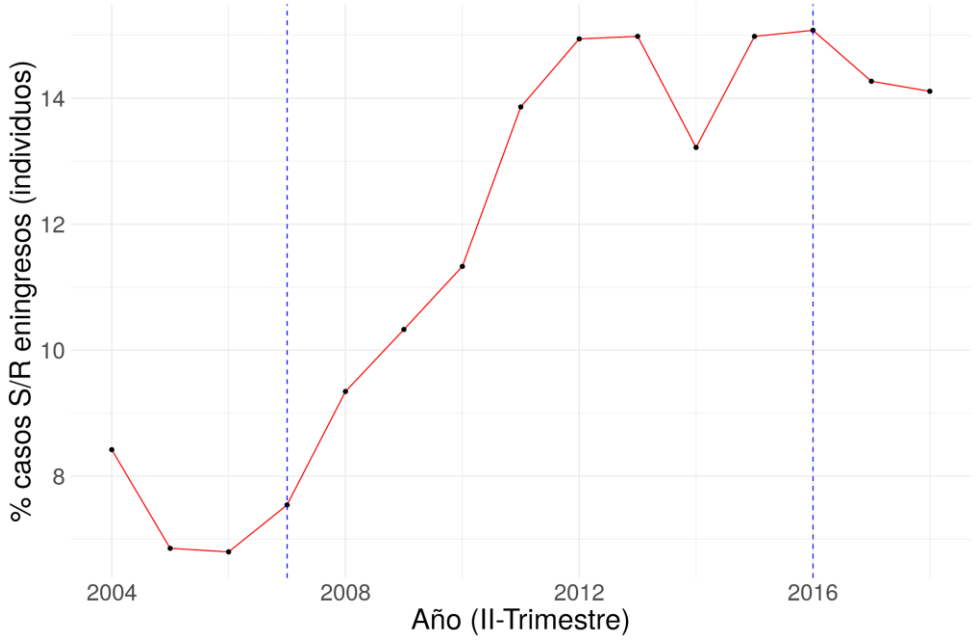
Este problema no es nuevo y, de hecho, en la EPH se observa un incremento notable en la proporción de valores de no respuesta totales, particularmente en dichas variables de ingreso. Diversos estudios (Salvia y Donza, 1999; Felcman, Kidyba y Ruffo, 2004; Pacífico, Jaccoud, Monteforte y Arakaki, 2011) parecen mostrar que la proporción de perceptores de ingresos con ingresos no declarados varió del 8% en 1995 al 24% en 2010 (luego de un descenso entre 1990 y 1994). El INDEC ha encarado el problema de diferentes formas. Durante la modalidad puntual de la EPH (hasta 2003), se optó por el método *pairwise*; luego, durante la primera etapa de la EPH-Continua se comenzó a utilizar el método *Hot Deck* combinado con la reponderación de ingresos; y, por último, desde 2016 se ha retomado el método de la reponderación (Camelo, 1999; Hoszowski, Messere y Tombolini, 2004; INDEC, 2009, 2017).

de la Universidad Nacional de Tres de Febrero. El equipo de trabajo se encuentra conformado, además del autor, por Hugo Delfino (codirector), María Giselle Galli, Adriana Chazarreta, Anabella Caputti y Julia Gentile (investigadores).

[4] Actualmente, se está trabajando para extender los alcances de los resultados a todo el período 2003-2015, en el que fue utilizado el método *Hot Deck*.

Recibido: 20/10/2020/Aceptado: 19/5/2021

Gráfico 1. Proporción de casos imputados (sin datos en alguna variable de ingresos) en EPH. Total de aglomerados urbanos, 2004-2018 (2°-trimestre de cada año).



Fuente: elaboración propia sobre microdatos de la EPH.

La magnitud del problema se pone de manifiesto al analizar la proporción de casos imputados (a nivel individuo) en alguna de las variables de ingreso en la Encuesta Permanente de Hogares entre 2004 y 2016. Se observa una tendencia creciente que comienza en 2006 y parece consolidarse a partir de 2007: pasa de menos de un 7% en 2006 hasta casi un 15% en 2015. A partir del 2016 la tendencia al incremento parece estancarse y mostrar un leve descenso, si bien queda en valores cercanos al 14%.

En la primera y segunda parte del artículo se plantea el problema de forma más específica y se pasa revista a los principales procesos teóricos de generación de valores perdidos, sus consecuencias, ventajas y limitaciones en la imputación de datos. En la tercera parte, se presentan las técnicas propuestas y sus fundamentos teóricos-metodológicos. Finalmente, en la cuarta sección, se presentan los principales resultados de la aplicación de los métodos propuestos sobre datos de la Encuesta Permanente de Hogares.

Recibido: 20/10/2020/Aceptado: 19/5/2021

Mecanismos de generación de datos perdidos y mecanismos habituales de resolución

En términos generales se asume que los datos perdidos y las no respuestas (NR) pueden ser generarse mediante tres procesos teóricos:

1) *Missing Completely at Random* (MCAR): en este caso, la probabilidad de que un registro tenga un valor perdido en la variable y no está relacionada con los valores de y ni con otros valores de la matriz de datos (Xs). Es decir, los datos perdidos son una submuestra aleatoria de la muestra general.

2) *Missing at Random* (MAR): si la probabilidad de NR en y es independiente de los valores de y, luego de condicionar sobre otras variables.

3) *Non Missing at Random* (NMR): en este caso, la probabilidad de NR depende tanto de variables Xs externas como de los valores de la variable con datos perdidos (y).

Resulta evidente que el supuesto MCAR no se cumple si: a) algún grupo o subgrupo tiene mayor probabilidad de presentar NR en la variable y; y/o b) si alguno de los valores de y tiene mayor probabilidad de NR.

El supuesto de MAR sería satisfecho si la probabilidad de no respuesta en ingresos dependiera, por ejemplo, exclusivamente del nivel educativo: es decir, si hubiera una mayor probabilidad en los niveles educativos altos de no haber respondido la variable ingresos. Bajo el proceso MAR, si bien existe esa probabilidad diferencial en cada grupo, al interior de cada uno de ellos (en el ejemplo anterior el nivel educativo) la probabilidad de no respuesta en ingresos no está relacionada con los valores del ingreso: dentro de los niveles educativos altos, todos los individuos tienen la misma probabilidad de presentar NR en la variable ingresos.

En términos generales, los datos perdidos no son generados por un proceso MAR si los casos con NR en una variable particular tienden a tener mayores o menores valores en esa variable que los casos con datos no perdidos, y se trataría de un proceso NMR.

Recibido: 20/10/2020/Aceptado: 19/5/2021

A grandes rasgos existen dos formas genéricas de lidiar con datos perdidos[5]: la primera es proceder a la eliminación de tales datos[6]. La segunda consiste en la imputación: se busca reemplazar los valores perdidos por una estimación razonable de ellos. A su vez, es posible identificar dos tipos de mecanismos de imputación: los basados en imputación simple y los basados en imputación múltiple. Entre los primeros podemos mencionar, por ejemplo, la imputación por media, por medias condicionadas, reponderación, etc.

Uno de los más utilizados métodos de imputación simple es el llamado *Hot Deck*. En este se busca reemplazar los valores perdidos de una o más variables de un caso no respondente (llamado "receptor") mediante los valores observados en un caso respondente (llamado "donante") similar al receptor. En algunas versiones el donante es seleccionado aleatoriamente de un set de potenciales donantes *(Random Hot Deck*); en otros casos se selecciona un solo caso donante, generalmente a partir de un algoritmo de "vecinos cercanos" y/o usando alguna métrica (*Deterministic Hot Deck*). En todos los casos, la imputación del valor perdido se realiza a partir de un solo valor estimado (Belin y Song, 2015).

Los métodos basados en las llamadas imputaciones múltiples (Van Buuren, 2018) generan un conjunto de posibles valores como estimación de los valores a imputar, los cuales son agregados de alguna manera. En general, se utilizan métodos de simulación de Monte Carlo y se sustituyen los datos faltantes a partir de un cierto número de simulaciones: "La metodología consta de varias etapas, y en cada simulación se analiza la matriz de datos completos a partir de métodos estadísticos convencionales y posteriormente se combinan los resultados para generar estimadores robustos" (Medina y Galván, 2007, p. 31).

Reseña de los métodos utilizados

Los dos primeros métodos explorados en este ejercicio presentan una lógica similar a la de imputación múltiple: generarán varias estimaciones para los valores perdidos a imputar, las cuales serán agregadas para generar el valor de imputación final. Se basan en un conjunto de técnicas englobadas bajo el nombre de *ensamble learning* (también llamadas ensamble de modelos, clasificadores basados en comités o sistemas de clasificadores múltiples). El objetivo

[5] En Rosati (2017) puede encontrarse una descripción más desarrollada de estos métodos.

[6] Existen dos formas de realizar esta eliminación: 1) exclusión *listwise*, en la que se trabaja solamente con los casos completos en toda la base; 2) exclusión *pairwise*: emplea solamente los datos completos de cada variable.

Recibido: 20/10/2020/Aceptado: 19/5/2021

general de los ensambles es incrementar la capacidad predictiva de clasificadores/modelos (*base learners*) a partir de la generación de submuestras de los datos originales y la estimación para cada una de esas submuestras de un modelo. Luego, las estimaciones provenientes de cada uno de esos modelos base se agregan de alguna manera y se genera la estimación final. De esta manera, se obtiene una capacidad predictiva que puede ser superior a la que presenta la aplicación de un solo clasificador base.

Los clasificadores base pueden ser de cualquier tipo (regresiones, árboles de clasificación, redes neuronales, etc.) e, incluso, puede plantearse la construcción de un ensamble con diferentes modelos base. Existen numerosos algoritmos para la construcción de ensambles de modelos y muchas aplicaciones a diversos problemas (Polikar, Zhang y Ma, 2012; Okun, Valentini y Re, 2011; Zhou, 2012). Pueden identificarse dos grandes metaalgoritmos de *ensamble learning*: *bagging* y *boosting*.

Bagging

Este meta-algoritmo (Breiman, 1996), que deriva su acrónimo del nombre **B**ootstrap **Agg**regat**ing**, simplemente entrena un determinado conjunto de clasificadores independientes, cada uno construido a través del remuestreo con reposición de los *n* registros del *training set*. La diversidad del ensamble se genera a partir de dos mecanismos: 1) la extracción de sucesivas muestras mediante el método *bootstrap*[7], y 2) la utilización de clasificadores débiles (sensibles a perturbaciones menores en los datos de entrenamiento). Algoritmos como los árboles de decisión –altamente sensibles– tienden a ser buenos candidatos para este propósito. Los clasificadores son combinados por alguna forma de voto mayoritario (medias, medianas, modas, etc.).

En este trabajo utilizaremos una versión particular del algoritmo de *bagging*, llamado *Random Forest* (Breiman, 2001; Hastie, Tibshirani y Friedman, 2015). Este algoritmo genera variabilidad tanto a nivel unidades de análisis (a través del *bootstrap*) como a nivel de los atributos. En cada iteración, no se utiliza la cantidad total de predictores (*M*) sino una muestra

[7] El *bootstrap* (Efron y Tibshirani, 1995) es un método consistente en extraer sucesivas remuestras con reposición de una muestra original. Es habitualmente utilizado para estimar distribuciones muestrales de diversos estimadores (medias, cuantiles, medidas de dispersión, ratios, etc.) y poder, de esta forma, realizar estimaciones de incerteza, por ejemplo.

Recibido: 20/10/2020/Aceptado: 19/5/2021

aleatoria de tal conjunto *M*: en lugar de particionar el espacio en función de los *M* predictores, para cada árbol se utiliza un subconjunto *m* de *M*. Cada árbol se construye a partir del siguiente algoritmo:

Se define un número total de árboles a entrenar (iteraciones).

Para cada iteración, a partir del número total de casos de entrenamiento (*n*) y del número de variables clasificadoras (*M*):

1. Se define el tamaño del subconjunto de *M* que se va a utilizar (*m*), tal que $m < M$.
2. Se extrae una muestra bootstrap de *n*.
    a. Para cada nodo del árbol:
        i. se muestrean *m* predictores de *M*,
        ii. se calcula cuál es la mejor partición a partir de las *m* variables del set de entrenamiento
    b. Cada árbol es desarrollado en su totalidad (es decir, no se realiza ningún tipo de "poda", como se haría en un clasificador simple basado en árboles).

Para la predicción se extrae una nueva muestra y se le asigna la etiqueta del nodo final del árbol. El procedimiento es iterado a lo largo de todos los árboles en el ensamble y cada caso es clasificado en la clase en que ha sido clasificado la mayor cantidad de veces a lo largo de todos los árboles generados (voto mayoritario).

Boosting

Los ensambles basados en *boosting*, también se basan en formas de remuestreo, aunque presentan algunas diferencias respecto a *bagging*. En cada nuevo paso, el algoritmo intentará corregir de alguna forma los errores cometidos en las iteraciones previas. De esta forma, trabaja sobre los errores del modelo entrenado en el paso anterior de dos formas posibles: 1) usándolos para cambiar la ponderación en el siguiente modelo o 2) entrenando un modelo que los prediga.

Así, en lugar de realizar un muestreo *bootstrap* de todos los datos indistintamente, en cada una de las iteraciones el algoritmo *boosting* se centra en aquellos registros en los que el clasificador funciona peor, es decir, en aquellos registros peor clasificados (Schapire y Freund, 2012). Se trata, entonces, de una construcción secuencial de cada uno de los estimadores base.

Recibido: 20/10/2020/Aceptado: 19/5/2021

En su versión más simple, llamada AdaBoost (acrónimo de **Ada**ptative **Boost**ing), se procede de la siguiente forma:

1. Se inicializan ponderaciones de todos los casos en $w_i = 1/n$.
2. Para cada iteración:
    a. se extrae una muestra del *dataset* con probabilidad w_i,
    b. se entrena un modelo (habitualmente un árbol de decisión),
    c. se calcula el error de clasificación ponderado del modelo (los casos mal clasificados pesan más en el error),
    d. se actualizan los pesos (w_i) en forma proporcional a la métrica de error.
3. Se agregan los resultados de cada iteración.

En este trabajo, utilizaremos un algoritmo llamado *Gradient Boosting* (Gerón, 2017) y particularmente una implementación optimizada llamada *XGBoost*, acrónimo de "e**X**treme **G**radient **Boost**ing", uno de los más utilizados en la actualidad. En lugar de modificar los pesos de los casos en el remuestreo, el algoritmo cambia de la siguiente forma:

1. Para cada iteración:
    a. Se entrena un modelo (ej., árbol de decisión),
    b. se calculan los residuos (error) del modelo,
    c. se entrena un modelo nuevo sobre los residuos,
    d. se agrega el nuevo modelo al ensamble.
2. Se agregan los resultados.

MultiLayer Perceptron (MLP)* o *Feed Forward Neural Network

El último método a utilizar no pertenece al conjunto de los clasificadores basados en ensambles: se trata de una red neuronal en una de sus arquitecturas más simples*: Multi-Layer Perceptron* (MLP), también llamada *Feed Forward Neural Network* (Goodfellow, Bengio, y Courville, 2018). Un MLP está compuesto de una capa de *input*, una capa de *output* y dos o más capas de unidades de cómputo denominadas LTU (*Linear Threshold Units*) o, más informalmente, "neuronas". Cada LTU toma como *input* un vector y realiza una transformación lineal generalmente seguida de una transformación no lineal. El valor resultante o *output* de esta transformación es enviado hacia la siguiente capa de la red. A medida que se "avanza" en

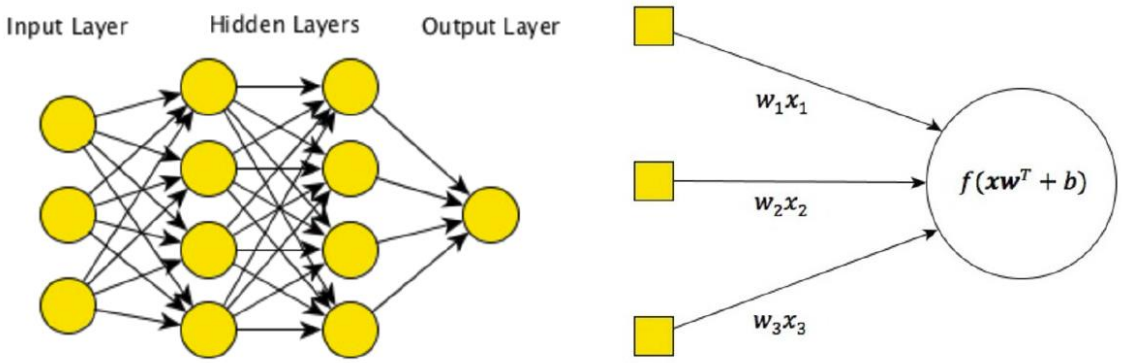
Recibido: 20/10/2020/Aceptado: 19/5/2021

la red, cada capa recibe como *input* el *output* de la capa previa. Existe una última capa que transforma el *input* en el formato deseado (números, clases, probabilidades, etc.[8]).

Una red neuronal, entonces, puede ser pensada como una composición de diferentes funciones encadenadas (Goodfellow, Bengio y Courville, 2018). Podríamos tener tres funciones encadenadas de la siguiente forma: $f(x) = f_3(f_2(f_1(x)))$. Así f_1 es la primera capa, f_2 es la segunda y f_3 es la última.

Veamos un esquema simple de un MLP, con dos capas ocultas o *hidden layers*, cada una de las cuales cuenta con 4 neuronas. A su vez, se observa una capa de entrada (que en nuestro caso, serán las diferentes variables predictoras, como sexo, edad, etc.) y una capa de salida, que consistirá en la predicción que la red hace de los ingresos de cada registro.

Esquema 1. *Multi-Layer Perceptron* y detalle de una neurona.



Fuente: Johan Edvinsson (2017). *Machine Learning at Condé Nast, Part 1: A Neural Network Primer* (https://technology.condenast.com/story/a-neural-network-primer).

En el ejemplo anterior, cada neurona de la primera capa recibe tres *inputs* (x_i), realiza una combinación lineal de cada uno ($w_i x_i + b$) y, por último, agrega el resultado. Es habitual que el *output* de una neurona sea transformado por una función de activación previamente a ser propagado hacia adelante en la red. De esta forma, el *output* final de una neurona es:

$$output = F_{activacion}\left(\sum_i x_i w_i^T + b\right)$$

[8] En este trabajo nos centramos solamente en las MLP en los que la información "fluye" en una sola dirección en la red. Existen arquitecturas de red (LSTM, convolucionales, etc.) más complejas que contienen *loops*, bucles, retroalimentaciones y saltos de información a lo largo de la red. Para mayor información sobre las diversas arquitecturas de red puede consultarse Gerón (2017) y Goodfellow, Bengio y Courville (2018).

Recibido: 20/10/2020/Aceptado: 19/5/2021

Existen diferentes funciones de activación, útiles para diferentes problemas. Las más comunes son las siguientes:

$$sigmoid : \sigma(x) = \frac{1}{1 + e^{-x}}$$
$$tanh : tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$
$$RELU : relu(x) = max(0, x)$$

El problema fundamental es, entonces, encontrar los parámetros de esta red[9]. En otras palabras, es necesario encontrar la combinación de pesos (w_i) que minimiza alguna función de pérdida. En el ejemplo del esquema 1 con 3 *inputs*, 2 capas de 4 neuronas cada una y un *output*, implica estimar 32 parámetros w_i. Es decir que el número de parámetros crece de forma no lineal con el tamaño del *dataset* y con el tamaño de la red.

Hiperparámetros y *overfitting*

Este punto nos lleva a una última cuestión: los hiperparámetros[10] fundamentales en un MLP son la cantidad de capas y la cantidad de unidades (o neuronas) que tendrá cada capa. Cuanta mayor cantidad de capas y/o neuronas posea la red, mayor será su complejidad: mayor será su capacidad para captar no linealidades e interacciones en los datos; pero, al mismo tiempo, mayor será también el riesgo de *overfitting*[11]. Esto vale también para los métodos *Random Forest* y *XGBoost*. Cada uno de ellos tiene un set de hiperparámetros a estimar: el

[9] Es fácil ver que un MLP con una sola capa, una sola neurona y sin función de activación es equivalente a una regresión lineal múltiple.

[10] En la terminología de *Machine Learning* y *Deep Learning*, los hiperparámetros son un conjunto de valores que deben ser seteados para controlar el comportamiento del algoritmo, principalmente para lidiar con el problema del *over* y *underfitting* (Goodfellow, Bengio y Courville, 2018). Generalmente, son estimados de forma iterativa mediante procedimientos de construcción de sets de validación (división entre datos de entrenamiento, datos de validación y datos de testeo, validación cruzada, etc.).

[11] El sobreajuste (u *overfitting*) se produce como consecuencia de sobreentrenar un algoritmo de aprendizaje automático o un modelo de predicción sobre un conjunto de datos sobre el que se conoce el valor de la variable que predecir. En general, se busca un modelo o algoritmo que logre una buena performance predictiva en datos nuevos, es decir, que permita generalizar la predicción a datos no observados previamente. Cuando se produce el sobreentrenamiento del modelo, existe la posibilidad de que el mismo ajuste "demasiado bien" a los datos de entrenamiento y, por ende, no capte la verdadera señal de los datos, sino que la confunda con ruidos y errores aleatorios de los datos. Como consecuencia, el modelo presenta un elevado ajuste en los datos de entrenamiento pero una mala performance en datos "nuevos". Existe amplia bibliografía al respecto, por ejemplo, ver Hastie, Tibshirani y Friedman (2009).

Recibido: 20/10/2020/Aceptado: 19/5/2021

primero, básicamente, la cantidad de árboles a entrenar (generalmente, denominado *m*). El segundo requiere la calibración de un conjunto más grande que van desde el *learning rate* hasta el grado de profundidad de cada uno de los árboles.

Los métodos propuestos tienen algunas ventajas en relación con las técnicas de imputación habitualmente utilizadas (específicamente, aquellas basadas en imputación simple o en la eliminación de casos). La construcción de ensambles de modelos introduce variabilidad en la estimación al remuestrear una determinada cantidad de veces los datos con valores a imputar. En efecto, esto permite potenciar la capacidad predictiva del modelo y generar clasificadores más eficientes. A su vez, el uso de una red neuronal permite el entrenamiento de modelos altamente no lineales en base a la combinación de transformaciones lineales en los datos.

Metodología de entrenamiento[12]

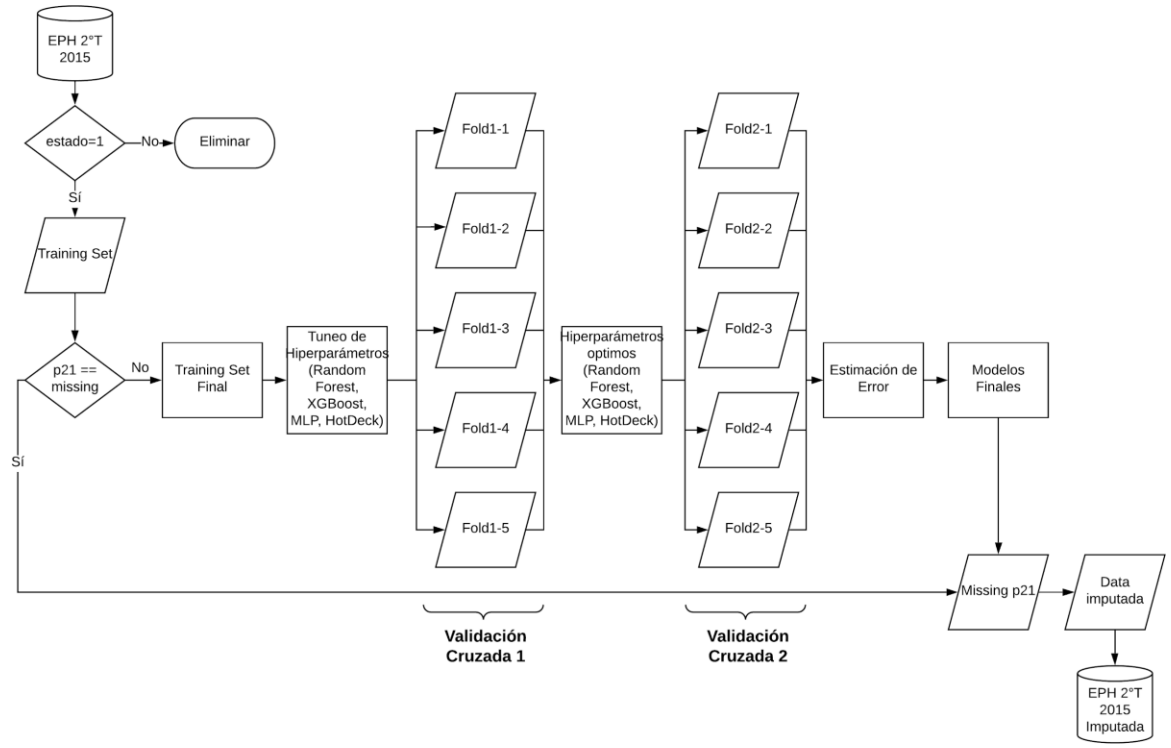
Los modelos fueron entrenados utilizando el lenguaje R y las librerías caret (para *XGBoost* y *RandomForest*), *Keras* (para el MLP) y hot.deck (para *Hot Deck*).

Para el entrenamiento de los modelos se utilizó la base usuaria de la Encuesta Permanente de Hogares correspondiente al segundo trimestre del año 2015. En todos los casos se procedió a la selección de los hiperparámetros de cada modelo mediante un esquema de validación cruzada de cinco grupos (k=5). Una vez seleccionados los mejores modelos (la mejor combinación de hiperparámetros para cada algoritmo) se procedió a dos formas de validación. Por un lado, se estimó el error de cada modelo mediante una nueva validación cruzada de k=5; dado que en este caso se particiona el *dataset* en 5 porciones y se estiman los errores en esta partición, esto equivale a la generación de datos perdidos de forma MAR o MCAR. En el siguiente esquema se resume el flujo de trabajo utilizado.

[12] El código y los datos para la replicación de los resultados puede encontrarse en https://github.com/gefero/ML_imputation.

Recibido: 20/10/2020/Aceptado: 19/5/2021

Esquema 2. Flujo de trabajo para la estimación y validación de los diferentes modelos.



Los datos de entrenamiento estuvieron constituidos por los ocupados de todas las categorías ocupacionales. Por otro lado, dado que se desconocen los parámetros y la forma concreta en que el INDEC realizó la imputación de valores, la segunda estrategia de validación consistió en comparar los datos imputados vía *Hot Deck* por el Instituto (identificables a partir del campo *IDIMPP*) con las imputaciones (sobre los mismos casos) realizadas por los tres métodos trabajados en esta ponencia.

En todos los casos, se utilizó como función de pérdida la raíz cuadrada del error cuadrático medio (ver más adelante) y se tomó como variable dependiente el monto de ingresos de la ocupación principal (p21) y como predictores las siguientes variables:

Recibido: 20/10/2020/Aceptado: 19/5/2021

Tabla 1. Predictores incluidos en los modelos

Variable	Dimensión
Región	Contexto
Aglomerado	
Tamaño	
Relación de parentesco	Sociodemográficas
Edad	
Sexo	
Situación conyugal	
Tipo de cobertura médica	
Sabe leer y escribir	
Nivel educativo	
Lugar de nacimiento	
Lugar de residencia	
Cantidad de ocupaciones	Ocupacionales
Total de horas trabajadas (semana de referencia)	
Intensidad de trabajo	
Búsqueda de mayor cantidad de horas de trabajo	
Categoría ocupacional	
Carácter de la ocupación	
Calificación de la ocupación	
Rama de actividad	
Tamaño del establecimiento	
Antigüedad en el empleo	
Cobertura previsional	
Percepción de ingresos por programas sociales	
Monto total de ingreso no laboral	

Recibido: 20/10/2020/Aceptado: 19/5/2021

A su vez, luego del proceso, los parámetros seleccionados para cada algoritmo fueron los siguientes:

Tabla 2. Especificaciones de cada modelo seleccionada

Algoritmo	**Parámetros óptimos**
Random Forest	{mtry=23, min.node.size=10}
XGBoost	{nrounds=200, max_depth=5, eta=0.1, gamma= 0.01, colsample_bytree=0.6, min_child_weight=0}
MLP	{layers=3; units=512; loss=mse; opt=rmsprop; activation=tanh; dropout=0.5}

Resultados

Se utilizarán dos indicadores habituales para la comparación de diferentes modelos de *Machine Learning* en problemas de regresión (es decir, con una variable dependiente de carácter cuantitativo, como es el caso de los ingresos de la ocupación principal). Para ello se compararán los valores observados (y_i) y los valores predichos ($\hat{y}_i$) para cada observación (i) realizada por cada modelo a partir de las siguientes métricas:

$$RMSE = \sqrt{\frac{\sum_i (y_i - \hat{y}_i)^2}{n}}$$

$$MAE = \frac{\sum_i |y_i - \hat{y_i}|}{n}$$

La primera fórmula corresponde al *Rooted Mean Squared Error* o la raíz cuadrada del Error Cuadrático Medio, es decir, la raíz del promedio de las diferencias entre valores observados y predichos. La segunda se llama *Mean Absolute Error* o Error Absoluto Medio, y mide la magnitud promedio de los errores (las diferencias entre valores esperados y observados) sin tomar en cuenta su signo.

Recibido: 20/10/2020/Aceptado: 19/5/2021

Dado que el RMSE se encuentra elevado al cuadrado, da una mayor ponderación a los errores de mayor magnitud. En cambio en el MAE todas las diferencias individuales tienen el mismo peso. Más allá de estas diferencias, ambas métricas cuantifican el error del modelo en la misma unidad que la variable a predecir (en este caso en pesos). A su vez, puede notarse de forma intuitiva que ambas decrecen a 0 cuando $y_i = \hat{y}_i$. De forma tal que un mejor ajuste del modelo (predicciones más parecidas a los valores observados) redundará en RMSE y MAE más pequeños.

Puesto que en este trabajo se utiliza un esquema de validación cruzada para evaluar los modelos seleccionados, las métricas a evaluar serán los promedios de cada una a lo largo de las k=5 particiones del *dataset*:

$$RMSE_{cv} = \frac{1}{K}\sum_{k=1}^{K}\frac{n_k}{n}RMSE_k$$

$$MAE_{cv} = \frac{1}{K}\sum_{k=1}^{K}\frac{n_k}{n}MAE_k$$

Así, cuando se comparen los diferentes modelos, se buscará detectar las mejoras en sus respectivas performances en relación a la magnitud en que logran reducir ambas métricas.

La tabla 3 muestra la capacidad que tienen los métodos de ensamble y de aprendizaje profundo para predecir valores perdidos siguiendo un patrón MCAR o MAR.

Tabla 3. Métricas de performance predictiva de los diferentes algoritmos entrenadas.

Algoritmo	RMSE	MAE
Hot Deck	$5,930.6	$3,740.6
Random Forest	$3,853.1	$2,263.0
XGBoost	$3,816.4	$2,248.0
MLP	$4,003.8	$2,310.9

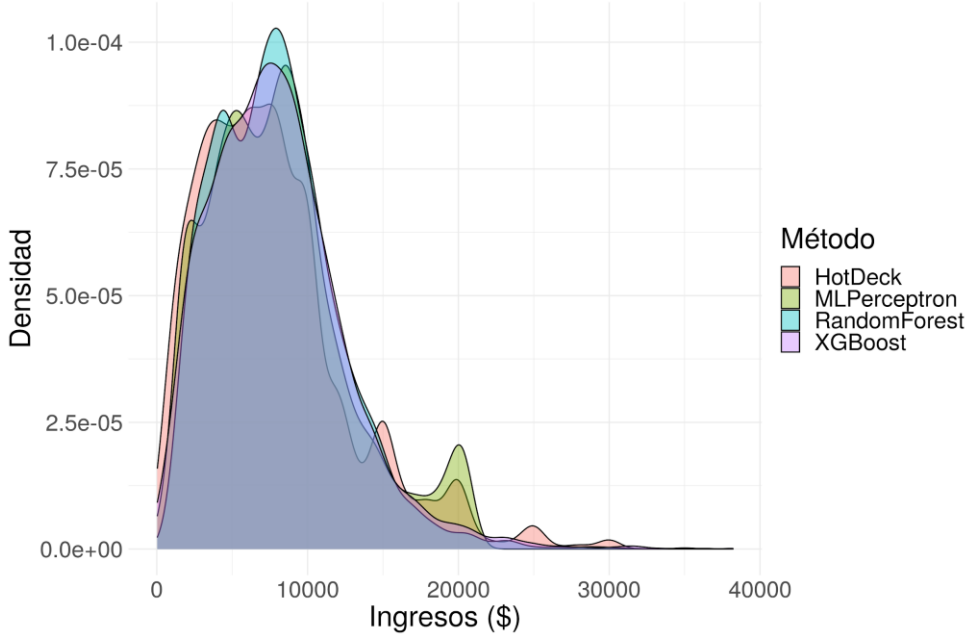
Fuente: elaboración propia en base a microdatos de la EPH-2.º trimestre de 2015.

Se observa que los métodos de ensamble superan ampliamente la performance de *Hot Deck*. Particularmente, si se observa el RMSE, se percibe que los tres métodos logran valores

alrededor de 33-35% inferiores a *Hot Deck*. Valores similares se obtienen al analizar el MAE. Estos resultados son relevantes dado que el método *Hot Deck* era utilizado por el INDEC para la imputación de valores perdidos en variables de ingreso en la EPH hasta fines del año 2015 y sigue siendo utilizado por la Dirección General de Estadísticas y Censos (DGEyC) de la Ciudad de Buenos Aires con los mismos fines en la Encuesta Anual de Hogares (EAH)[13].

Ahora bien, ¿qué impacto tienen las predicciones realizadas por estos modelos en la distribución de los valores imputados? ¿Cambian sustancialmente las estimaciones de ingresos según se realice una imputación vía *Hot Deck* o alguno de estos otros métodos? Para realizar una primera aproximación a estas preguntas se centrará la mirada en los datos imputados originalmente por el INDEC (usando el método *Hot Deck*) y se realizarán imputaciones alternativas mediante cada uno de los métodos analizados.

Gráfico 2. *Density plot* de la variable ingresos de la ocupación principal (p21) por método de imputación (casos imputados).



Fuente: elaboración propia en base a microdatos de la EPH-2.º trimestre de 2015.

[13] Desde 2012 se publican los montos de ingresos imputados en la base usuaria de la EAH. Según la DGEyC (2016: p. 46), "el método de imputación ingresos empleado en la EAH se conoce como procedimiento hot-deck jerárquico secuencial. Este pertenece a los métodos que emplean 'donantes' y 'receptores'. Los registros 'donantes' son aquellos registros completos de la encuesta que se emplean para asignar su valor, mientras que los registros 'receptores o candidatos' son los que tienen valores faltantes, es decir reciben un valor". También Manzano (2016) menciona que "se probaron varios métodos de imputación y finalmente se optó por imputar los ingresos a nivel de fuente de ingreso, seleccionando donantes con el método hot-deck jerárquico, dentro de clases de imputación construidas utilizando árboles de regresión y regresiones categóricas". Para mayor información sobre el método *Hot Deck* puede consultarse Belin y Song (2015).

Recibido: 20/10/2020/Aceptado: 19/5/2021

Puede verse en el gráfico anterior que *Hot Deck* parece ser uno de los métodos con mayor ruido en los valores extremos de la distribución. En efecto, en ambas colas de la distribución se observa una mayor densidad de casos. Algo similar, aunque en menor medida, se observa en el caso de MLP. En cambio, los métodos de *Random Forest* y *XGBoost* parecen ser los más suaves y con menor dispersión en este sentido.

Tabla 4. Estadísticos descriptivos de la variable ingresos de la ocupación principal (p21) según método de imputación (casos imputados).

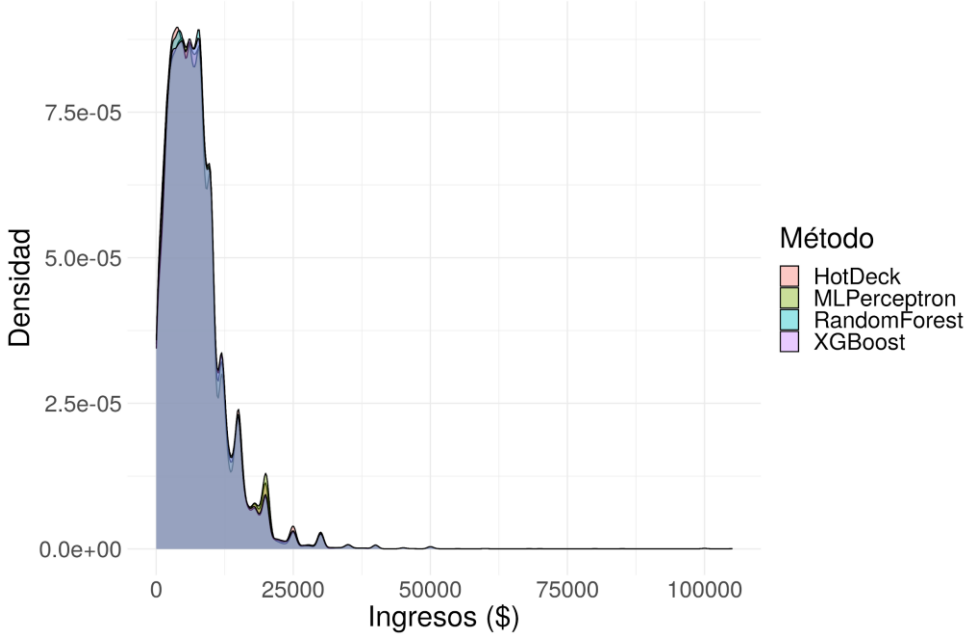
Algoritmo	Min	Q1	Q2	Media	Q3	Max	CV
Hot Deck	$200.0	$4,000.0	$7,000.0	$7,729.0	$10,000.0	$35,000.0	66.4%
MLPerceptron	$111.2	$4,622.7	$7,619.1	$8,048.5	$10,289.3	$20,678.5	57.2%
Random Forest	$740.6	$4,689.0	$7,570.3	$7,885.2	$10,215.1	$36,288.7	52.5%
XGBoost	$37.1	$4,722.1	$7,504.3	$7,940.0	$10,345.39	$38,206.2	56.0%

Fuente: elaboración propia en base a microdatos de la EPH-2.º trimestre de 2015.

Los descriptivos anteriores confirman la imagen del gráfico: similitud en el centro de la distribución y divergencia en las colas. A su vez, la dispersión es elevada en todos los métodos, pero lo es en mayor medida en el caso de *Hot Deck*.

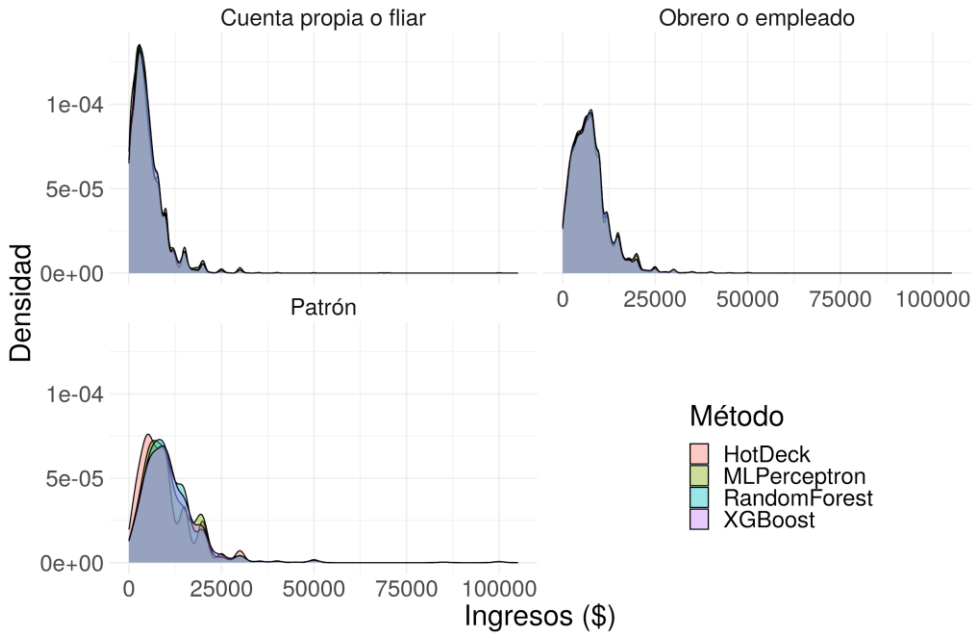
Recibido: 20/10/2020/Aceptado: 19/5/2021

Gráfico 3. *Density plot* de la variable ingresos de la ocupación principal (p21) por método de imputación (casos imputados y completos).



Fuente: elaboración propia en base a microdatos de la EPH-2.º trimestre de 2015.

Gráfico 4. *Density plot* de la variable ingresos de la ocupación principal (p21) por método de imputación según categoría ocupacional (casos imputados y completos).



Fuente: elaboración propia en base a microdatos de la EPH-2.º trimestre de 2015.

Recibido: 20/10/2020/Aceptado: 19/5/2021

Al comparar, ahora, la distribución total de la variable (es decir, casos imputados y completos), esta no parece alterarse sustancialmente. Algo similar sucede si se analiza la distribución condicionada a la categoría ocupacional. Entre asalariados y trabajadores por cuenta propia o familiares no hay diferencias notables. Solamente se observan algunas diferencias en los patrones.

Discusión

En el presente trabajo se presentaron los resultados de algunos experimentos para la estimación de modelos de imputación de datos perdidos utilizando técnicas de *Machine Learning*: ensambles y perceptrones multicapa. Luego de exponer los fundamentos generales de cada una de las técnicas se realizaron dos tipos de evaluación de los modelos con base en los datos de la EPH (2do trimestre de 2015) buscando lograr la imputación de la variable correspondiente a los ingresos de la ocupación principal de los individuos.

Se mostró la mayor performance que los ensambles y el MLP tienen en comparación con la técnica habitualmente utilizada en algunas dependencias del Sistema Estadístico Nacional. En efecto, al cuantificar dos indicadores usuales para este tipo de problemas se observó que el RMSE de los modelos basados en *Machine Learning* oscilaba alrededor de los $3.800 y $4.000, mientras que el RMSE de *Hot Deck* superaba los $5.900, lo cual implica una mejora de alrededor del 33%. Valores similares mostraba el indicador MAE.

A su vez se mostró que las distribuciones en las imputaciones basadas en cada método muestran similitudes en el centro y algunas diferencias en las colas, siendo la de *Hot Deck* la más inestable: parece comportarse más erráticamente en los extremos de la distribución (es decir, en los casos de mayores y menores ingresos). A su vez, al observar la distribución completa de la variable ingresos (casos completos e imputados) se observan escasas diferencias en cada uno de los métodos. Esto se mantiene al condicionar según la categoría ocupacional.

Ahora bien, se abren una serie de líneas de trabajo a explorar en futuras aproximaciones. La primera de ellas tiene que ver con extender el alcance del ejercicio en dos direcciones: por un lado, incorporar mayor cantidad de información proveniente de la EPH (diferentes años y trimestres) para evaluar los modelos analizados en este trabajo; por otro, incorporar otras encuestas a hogares, incluso de otros países. También resulta relevante estudiar las propiedades de los estimadores y las estimaciones al utilizar uno u otro método de imputación: ¿cómo varían

Recibido: 20/10/2020/Aceptado: 19/5/2021

las estimaciones de indicadores clásicos basados en los ingresos como, por ejemplo, el coeficiente de Gini o las estimaciones de pobreza? ¿Qué sucede con los intervalos de confianza de tales estimadores? ¿Qué efecto tiene cada uno de los métodos de imputación sobre los valores de estos indicadores?

A su vez, específicamente en términos de los modelos analizados, queda pendiente realizar entrenamientos con grillas de hiperparámetros más exhaustivas y, dado que el tiempo de cómputo es una restricción para considerar, evaluar algoritmos de optimización más inteligentes que una búsqueda por fuerza bruta (algoritmos genéticos, optimización bayesiana). Particularmente, en el caso del MLP, será necesario probar otras combinaciones de hiperparámetros (diferentes funciones de activación, tasas de *dropout*, funciones de pérdida más robustas, etc.) e incluso otras arquitecturas de redes neuronales más complejas (como redes convolucionales o residuales).

Finalmente, en este trabajo se intentó mostrar que las técnicas analizadas tienden a mejorar significativamente la performance de métodos simples como *Hot Deck*. Sin embargo, en este ejercicio se asumió un proceso generador de datos perdidos MCAR o MAR. Queda pendiente evaluar la performance relativa comparada con *Hot Deck* para procesos de generación no aleatorios[14].

Referencias bibliográficas

Belin, T. y Song, J. (2015). "Missing data in survey analysis". En Molenberghs, G., Fitzmaurice, G., Kenward, M., Tsiatis, A. y Verbeke, G. (eds.). *Handbook of Missing Data Methodology*. New York: Taylor & Francis/CRC: 525-546.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24, 123-140.

—— (2001). Random Forest. *Machine Learning*, 42, 5-32.

Camelo, H. (1999). *Subdeclaración de ingresos medios en las encuestas de hogares, según quintiles de hogares y fuente del ingreso.* Trabajo presentado en II Taller Programa para el Mejoramiento de las Encuestas y la Medición de las Condiciones de Vida en América Latina y el Caribe (MECOVI), Buenos Aires.

Dirección General de Estadísticas y Censos (2016). *Base Usuarios Ampliada 2015. Encuesta Anual de Hogares de la Ciudad de Buenos Aires*. Disponible en https://www.estadisticaciudad.gob.ar/eyc/?cat=93.

[14] María Giselle Galli, miembro del equipo, se encuentra desarrollando su tesis de maestría en esta dirección.

Efron, B. y Tibshirani, R. (1995). *An Introduction to the Bootstrap*. Florida: Chapman & Hall/CRC.

Felcman, D., Kidyba, S. y Ruffo, H. (2004). *Medición del ingreso laboral: ajustes a los datos de la encuesta permanente de hogares para el análisis de la distribución del ingreso (1993–2002)*. Trabajo presentado en el XIV Taller Programa para el Mejoramiento de las Encuestas y la Medición de las Condiciones de Vida en América Latina y el Caribe (MECOVI), Buenos Aires.

Gerón, A. (2017). *Hands-on Machine Learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. Boston: O'Reilly.

Goodfellow, I., Bengio, Y. y Courville, A. (2018). *Deep Learning*. Boston: MIT Press.

Hastie, T., Tibshirani, R. y Friedman, J. (2009). *The Elements of Statistical Learning. Data Mining, Inference, and Prediction.* Berlin: Springer.

Hoszowski, A., Messere, M., y Tombolini, L. (2004). *Tratamiento de la no respuesta a las variables de ingreso en la Encuesta Permanente de Hogares de Argentina*. Trabajo presentado en el XIV Taller Programa para el Mejoramiento de las Encuestas y la Medición de las Condiciones de Vida en América Latina y el Caribe (MECOVI), Buenos Aires.

INDEC (2009). *Ponderación de la muestra y tratamiento de valores faltantes en las variables de ingreso en la EPH*. Metodología N.° 15. Buenos Aires.

—— (2017). *No respuesta de ingresos en la Encuesta Permanente de Hogares*. Documento Técnico (sin número), INDEC, Buenos Aires. Disponible en: https://www.indec.gob.ar/ftp/cuadros/sociedad/nota_EPH_ingresos_06_17.pdf.

Manzano, G. (2016). *Imputación de datos de ingresos en encuestas a hogares. La experiencia de la Encuesta Anual De Hogares (EAH) de la Dirección General de Estadística y Censos de la Ciudad De Buenos Aires de Argentina*. Resumen de ponencia presentada en el 3.º ISA Forum of Sociology. Disponible en https://isaconf.confex.com/isaconf/forum2016/webprogram/Paper78501.html.

Medina, F. y Galván, M. (2007). Imputación de datos: teoría y práctica". Serie Estudios Estadísticos y Prospectivos, 54, Santiago de Chile: CEPAL. Disponible en http://www.cepal.org/es/publicaciones/4755-imputacion-datos-teoriapractica.

Okun, O., Valentini, G. y Re, M. (2011). *Ensambles in Machine Learning Applications*. Berlín: Springer.

Pacífico, L., Jaccoud, F., Monteforte, E., y Arakaki, G.A. (2011). *La Encuesta Permanente de Hogares, 2003-2010. Un análisis de los efectos de los cambios metodológicos sobre los principales indicadores sociales*. Trabajo presentado en el X Congreso Nacional de Estudios del Trabajo, Buenos Aires.

Polikar, R., Zhang, C., y Ma, Y. (eds.) (2012). *Ensamble Machine Learning. Methods and Applications*. Berlín: Springer.

Recibido: 20/10/2020/Aceptado: 19/5/2021

Rosati, G. (2017). Construcción de un modelo de imputación para variables de ingreso con valores perdidos a partir de ensamble learning. Aplicación a la Encuesta Permanente de Hogares. *Revista Saberes*, 9(1), 91-111.

Salganik, M. (2018). *Bit by bit. Social research in the digital age*. New Jersey: Princeton University Press.

Salvia, A. y Donza, E. (1999). Problemas de medición y sesgos de estimación derivados de la no respuesta completa a las preguntas de ingresos en la EPH (1990-1998). *Estudios del Trabajo*, 18, 93-110.

Schapire, R. y Freund, Y. (2012). *Boosting: Foundations and Algorithms* . *Massachussets*: MIT Press.

Van Buuren, S. (2018). *Flexible Imputation of Missing Data*. New York: Taylor & Francis/CRC.

Zhou, Z. (2012). *Ensamble Methods. Foundations and Algorithms*. Florida: Chapman & Hall/CRC.