ASOCIACIÓN ARGENTINA DE ESPECIALISTAS EN ESTUDIOS DEL TRABAJO
CONGRESO NACIONAL DE ESTUDIOS DEL TRABAJO
LOS TRABAJADORES Y LAS TRABAJADORAS EN EL ESCENARIO ACTUAL.
Condiciones estructurales y alternativas frente a la crisis
BUENOS AIRES, 7, 8 Y 9 DE AGOSTO DE 2019

Grupo Temático N° 20: Abordajes conceptuales y metodológicos en torno a las temáticas asociadas a los estudios del trabajo

Coordinadores: Cynthia Pok, María Albina Pol, Andrea Lorenzetti

---

Imputación de datos perdidos mediante técnicas de Machine Learning: un experimento usando la Encuesta Permanente de Hogares

Autor/a: Germán Rosati

E–mail: german.rosati@gmail.com

Pertenencia institucional: IDAES-UNSAM/ CONICET/ PIMSA

1. Introducción

La presencia de no respuesta y valores perdidos representan un problema recurrente e histórico 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 estadístico. 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, etc.) se caracterizan por formatos no estructurados, orígenes diversos e inconsistencias varias. La expansión en la utilización de las mismas muestra la necesidad de contar con herramientas que permitan lidiar con la existencia de datos perdidos de forma performante.

Este punto tiene particular relevancia dado que "las rutinas de los paquetes estadísticos asumen que se trabaja con datos completos incorporan opciones –no siempre las más adecuadas– para imputar observaciones sin que el usuario se dé cuenta de ello. […] la aplicación de procedimientos inapropiados […] introduce sesgos y reduce el poder explicativo de los métodos estadísticos" (Medina y Galvan, 2007, p.10).

La presente ponencia expone algunos avances iniciales en la evaluación de diferentes modelos para la imputación de valores perdidos y sin respuesta para las variables de ingreso en encuestas a hogares.

Continúa una línea de trabajo enmarcada en un proyecto más general[1] que busca evaluar la capacidad de las técnicas de Machine Learning para la imputación de datos perdidos en variables cuantitativas y cualitativas.

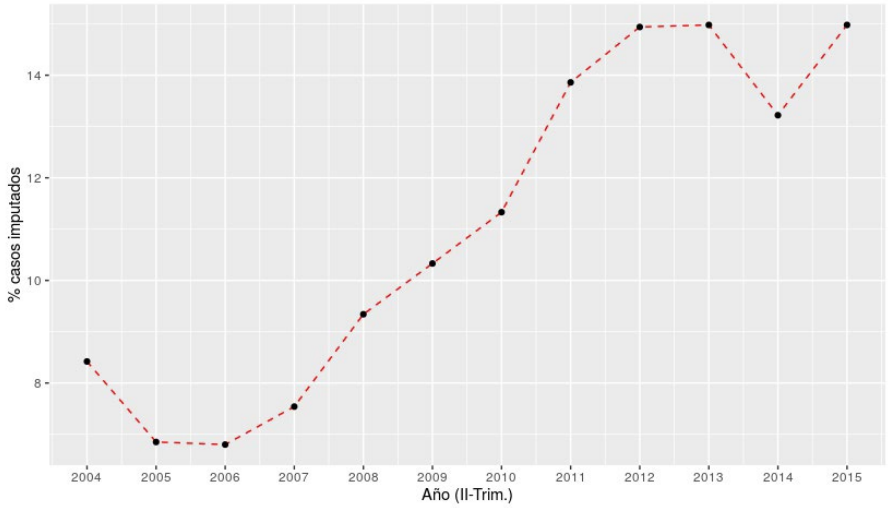
Se presentan los resultados preliminares de algunos experimentos de imputación de la variable de ingresos laborales de la Encuesta Permanente de Hogares, basados en técnicas de Ensamble Learning y Deep Learning: Random Forest, XGBoost y Multi-Layer Perceptron. Se compara la performance de estas técnicas con el método Hot Deck (uno de los métodos usados por el Sistema Estadístico Nacional).

Los métodos explorados resultan de carácter general y, por tanto, 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 en la región) se presenta aquí una aplicación para la imputación de variables de ingresos utilizando microdatos correspondientes al 2do. trimestre de 2015 de la Encuesta Permanente de Hogares (EPH) relevamiento elaborado por el Instituto Nacional de Estadísticas y Censos de la Argentina (INDEC).

El problema no es nuevo y, de hecho, la EPH ha visto incrementarse la proporción de valores de no respuesta totales particularmente en variables de ingreso. Los 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). En ese sentido, el INDEC ha encarado el problema de diferentes formas en la EPH. Durante la EPH en su modalidad puntual, 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, a finales de 2015 se ha retomado el método de la reponderación (Camelo, 1999; Hoszowski, Messere y Tombolini, 2004; INDEC, 2009).

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

Gráfico 1. Proporción de casos imputados (sin datos en alguna variable de ingresos) en EPH. Total de aglomerados urbanos, 2003-2015 (II-Trimestre de cada año).



Fuente: elaboración propia sobre microdatos EPH

La magnitud del problema se pone de manifiesto al analizar la proporción de casos imputados (a nivel invididuo) en alguna de las variables de ingreso en la Encuesta Permanente de Hogares entre 2004 y 2015. Se observa una tendencia creciente que va desde un 8,5% en 2004 hasta casi un 15% en 2015.

En la primera y segunda parte del documento 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.

2. Mecanismos de generación de datos perdidos y mecanismos habituales de resolución

En términos generales se asume que los datos perdidos se generan mediante tres procesos:

> *1) Missing Completly 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 (X´s). Es decir, los datos perdidos son una submuestra aleatoria de la muestra general. Este supuesto se viola si: a) algún grupo o subgrupo tiene mayor

probabilidad de presentar NR (no respuesta) en la variable Y; y/o b) si alguno de los valores de Y tiene mayor probabilidad de NR.

2) Missing at Random (MAR): si la probabilidad de NR en Y es independiente de los valores de Y, por lo tanto de condicionar sobre otras variables.

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

El supuesto de MAR sería satisfecho si la probabilidad de no respuesta en ingresos dependiera de, por ejemplo, el 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.

En general, existen dos grandes formas de lidiar con datos perdidos[2]: la primera de ellas es eliminar tales casos[3]. La segunda consiste en la imputación: se busca reemplazar los valores perdidos por una estimación razonable de los mismos. Es posible identificar dos grandes tipos de mecanismos de imputación: los basados en "imputación simple" y los basados en "imputación múltiple". Entre los primeros podemos mencionar la imputación por media, por medias condicionadas, reponderación.

Uno de los más utilizados es el llamado método Hot Deck, en el cual se busca reemplazar los valores perdidos de una o más variables de un no respondente ("receptor") con los valores observados de un respondente ("donante") que es 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" usando alguna métrica (deterministic hot deck). En todos, la imputación del valor perdido se realiza a partir de un solo valor estimado. Los métodos basados en las llamadas "imputaciones múltiples" generan un conjunto de posibles valores como estimación de los valores a imputar, los cuales son agregados de alguna manera.

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

[3] 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.

En general, se utilizan métodos de simulación de Monte Carlo y se sustituyen los datos faltantes a partir de un número (mayor a 1) 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).

Los métodos propuestos en este documento se basan en una lógica similar a la de imputación múltiple: generarán varias estimaciones para los valores perdidos a imputar y las agregarán para generar el valor de imputación final

3. Reseña de los métodos utilizados

Los dos primeros métodos explorados en este ejercicio 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 general de los ensambles de modelos 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 generados se agregan de alguna manera y se obtiene 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.

Este *base learner* puede 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 meta-algoritmos de *ensamble learning*, es decir, dos grande estrategias para la *bagging y boosting*.

Bagging

El primero (Breiman, 1996) -cuyo nombre deriva del acrónimo **B**ootstrap **AGG**regat**ING**) simplemente entrena un determinado conjunto de clasificadores independientes, cada uno construido a

través del remuestreo con reposición de *n* registros del *training set*. La diversidad del ensamble está asegurada por la variación de las muestras *bootstrap* y por la utilización de clasificadores débiles (sensibles a perturbaciones menores en los datos de entrenamiento). En ese sentido, algoritmos como los árboles de decisión tienden a ser buenos candidatos para este propósito. Los clasificadores son combinados por alguna forma de simple majority voting (medias, medianas, modas, etc.).

En este trabajo utilizaremos una versión particular del algoritmo de *bagging*, el llamado *Random Forest* (Breiman 2001, Hastie, Tibshirani, et al 2015). En este algoritmo se genera variabilidad tanto a nivel filas (*bootstrap*) como a nivel columnas. En cada iteración, no se utiliza la cantidad total de predictores (*M*) sino que se utiliza una muestra aleatoria de *M*: no se particiona el espacio de predictores en función de los *M* predictores, sino que para cada árbol, se utiliza un subconjunto *m* de *M*. Cada árbol se construye a partir del siguiente algoritmo:

1. A partir del número total de casos de entrenamiento (*D*) y del número de variables clasificadoras (*M*),
    a. definir el tamaño del subconjunto de *M* que se va a utilizar (*m*), tal que $m < M$
    b. extraer una muestra *bootstrap* de *D*
    c. Para cada nodo del árbol
        i. muestrear *m* predictores de *M*
        ii. calcular cuál es la mejor partición a partir de las *m* variables del set de entrenamiento
    d. Cada árbol es desarrollado en su totalidad (es decir, no se realiza ningún tipo de "poda", como se haría en un clasificador basado en árboles habitual).

Para la predicción una nueva muestra es extraída. 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* presentan, si bien también se basan en diferentes formas de remuestreo, algunas diferencias respecto al *bagging*. En cada nuevo paso el algoritmo intentará

aprender de los errores cometidos en los pasos previos. Trabaja sobre los errores del modelo anterior o bien usándolos para cambiar la ponderación en el siguiente modelo o bien entrenando un modelo que prediga los mismos. Así, en lugar de realizar un muestreo *bootstrap* de todos los datos indistintamente, en cada una de las iteraciones el algoritmo 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. En su versión más simple, llamada *AdaBoost*, se procede de la siguiente forma:

1. Se inicializan las ponderaciones de todos los casos en wi=1/n
2. Para cada iteración (en cada una se entrena un árbol de decisión):
    a. se extrae una muestra del dataset con probabilidad wi
    b. se entrena un modelo (por ejemplo, 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
1. Se agregan los resultados de cada iteración

En este trabajo, utilizaremos un algoritmo llamado Gradient Boosting -Gerón 2017- (particularmente, una implementación 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:

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

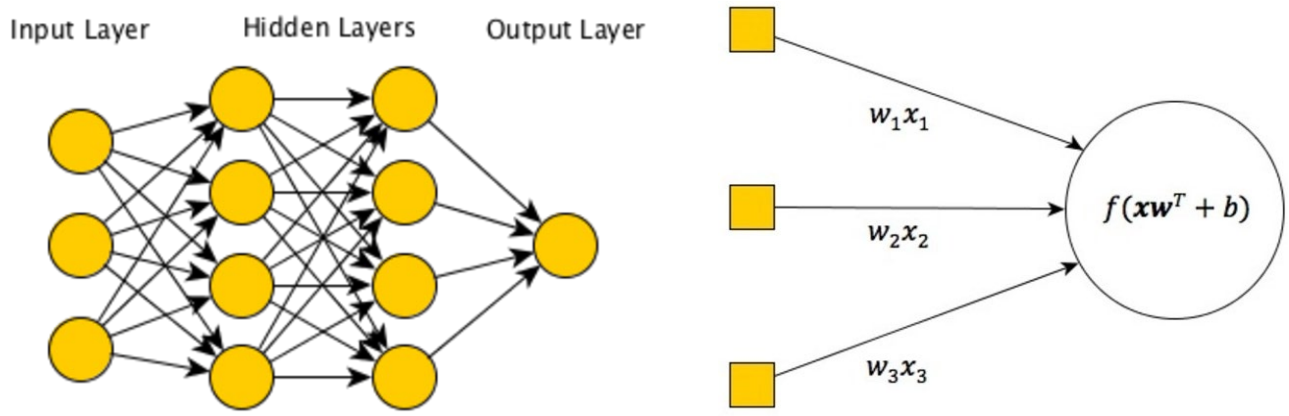
Multi Layer Perceptron (MLP) o Feed Forward Neural Network

El último método a utilizar será una red neuronal, particularmente en una de sus arquitecturas más simples: multilayer perceptron, también llamada feed forward neural network (Goodfellow, Bengio, y Courville 2016). Las redes neuronales son una familia de algoritmos compuestos de "unidades de

cómputo" llamadas neuronas. Cada neurona toma un vector como input y realiza alguna forma de combinación lineal, a menudo seguida de alguna función no lineal. El valor resultante de esta transformación es "propagado" hacia adelante en la red. El input de las primeras capas de neuronas es el dataset crudo, y a medida que se avanza en la red, cada capa recibe el output de la capa previa. Una capa final transforma el input en el formato de output deseado (números, clases, probabilidades, etc[4].

Veamos un esquema simple de un MLP, el mismo tiene dos capas ocultas o hidden layers, cada una de las cuales, cuenta con 4 neuronas. A su vez, se observa una capa de input (que en nuestro caso, serán las diferentes variables predictoras (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: https://technology.condenast.com/story/a-neural-network-primer

La unidad fundamental de una red neuronal es la llamada "neurona", veamos qué sucede dentro de una neurona. La misma recibe tres inputs (x_i) y realiza una combinación lineal de cada input ($w_ix_i + b$) y agrega el resultado utilizando alguna función f(). Es habitual, además que el output de una neurona sea transformado por lo que se llama "función de activación" previamente a ser propagado hacia adelante en la red. De esta forma, el output final de una neurona es:

[4] 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 en las existen loops, bucles, retroalimentaciones y saltos de información a lo largo de la red. Para mayor información sobre las diversas arquitecturas de red puede consultarse Geron (2017) y Goodfellow, Bengio y Courville, A. (2017).

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

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

$$sigmoid: \quad \sigma(x) = \frac{1}{1+e^{-x}}$$

$$tanh: \quad \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$RELU: \quad relu(x) = \max(0, x)$$

El problema fundamental a resolver, es entonces, encontrar los parámetros de esta red[5]. Es decir, es necesario encontrar la combinación de pesos w_i que minimiza alguna función de pérdida. En este caso "de juguete" 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.

Este punto nos lleva a una última cuestión: los hiperparámetros fundamentales en un MLP son la cantidad de capas y la cantidad de unidades (o neuronas) que tendrá cada capa. Cuanto mayor cantidad de capas y/o neuronas posea la read, tendrá una complejidad superior, es decir, mayor será su capacidad para captar no linealidades e interacciones en los datos. Pero, al mismo tiempo -y esto vale para el resto de los modelos analizados- mayor será el riesgo de overfitting[6].

Los métodos propuestos tienen algunas ventajas en relación a las técnicas de imputación habitualmente utilizadas (específicamente, aquellas basadas en imputación simple o en la eliminación –listwise o pairwise– de casos). La construcción de ensambles de modelos introduce variabilidad en la estimación al remuestrar 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,

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

[6] 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 a predecir. En general, se busca un modelo o algoritmo que logra 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).

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.

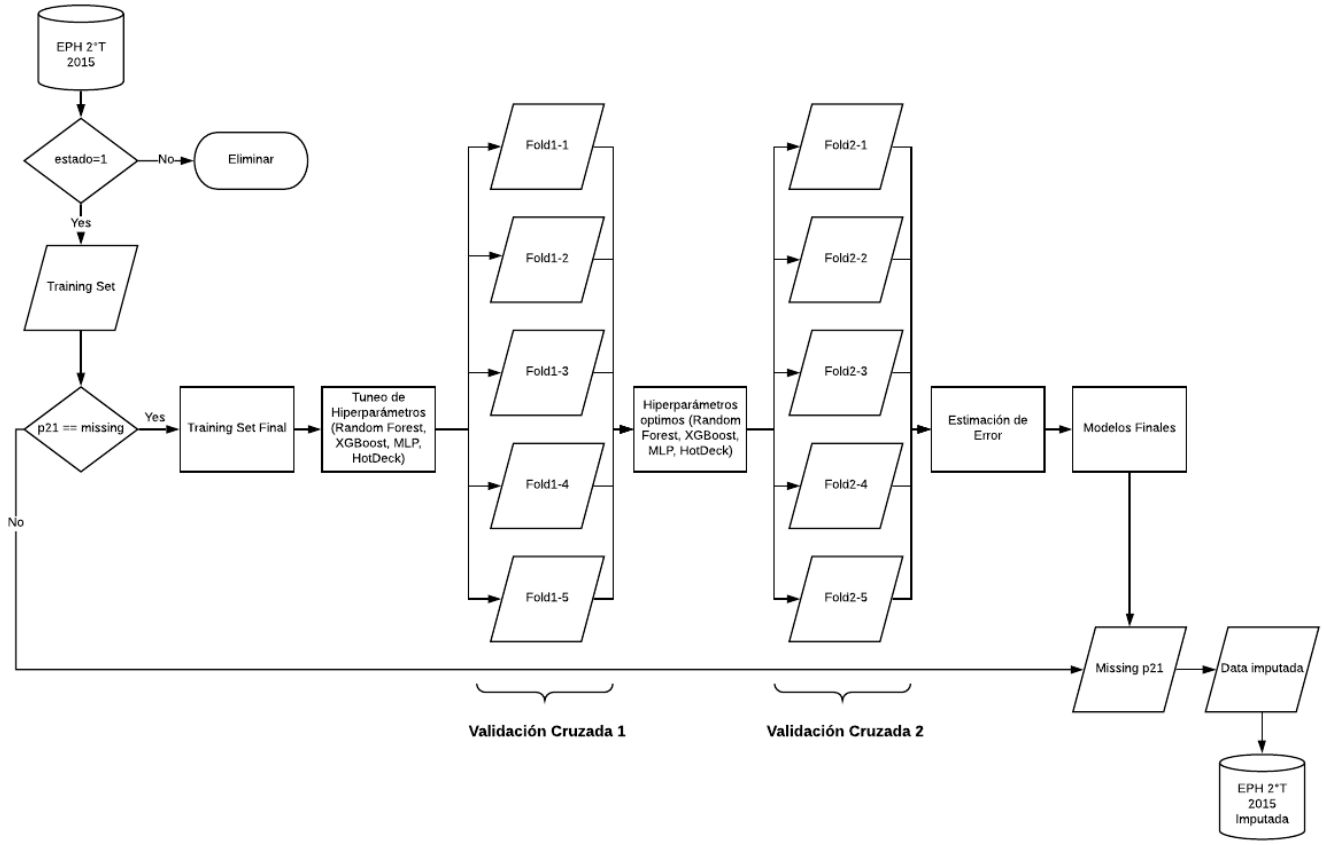
4. Metodología de entrenamiento[7]

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 a 3er. trimestre del año 2015. En todos los casos se procedió al tuneo de los hiperparámetros de cada modelo mediante un esquema de validación cruzada de k=5. Una vez seleccionado los mejores modelos (la mejor combinación de hiperparámetros para cada algoritmo) se procedió a dos formas de validación. En primer lugar, 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.

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

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



Dado que se desconoce 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 INDEC, identificables a partir del campo iidimp) con las imputaciones (sobre los mismos casos) realizadas por los tres métodos trabajados en esta ponencia. En todos los casos, se minimizó el error cuadrático medio y se tomó como variable dependiente el monto de ingresos laborales (p21) y como predictores las siguientes variables:

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	

A su vez, luego del proceso, los parámetros seleccionados para cada algoritmo son 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}

5. Resultados

Performance predictiva bajo supuestos MAR o MCAR

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

Algoritmo	**RMSE**	**MAE**
Hot Deck	$5930.6	$3740.6
Random Forest	$2800.6	$1561.9
XGBoost	$3260.8	$2016.8
MLP	$3974.2	$2293.1

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

Estos resultados (preliminares) muestran la capacidad que tienen los métodos de ensamble y de aprendizaje profundo para predecir valores perdidos siguiendo un patrón MCAR o MAR.

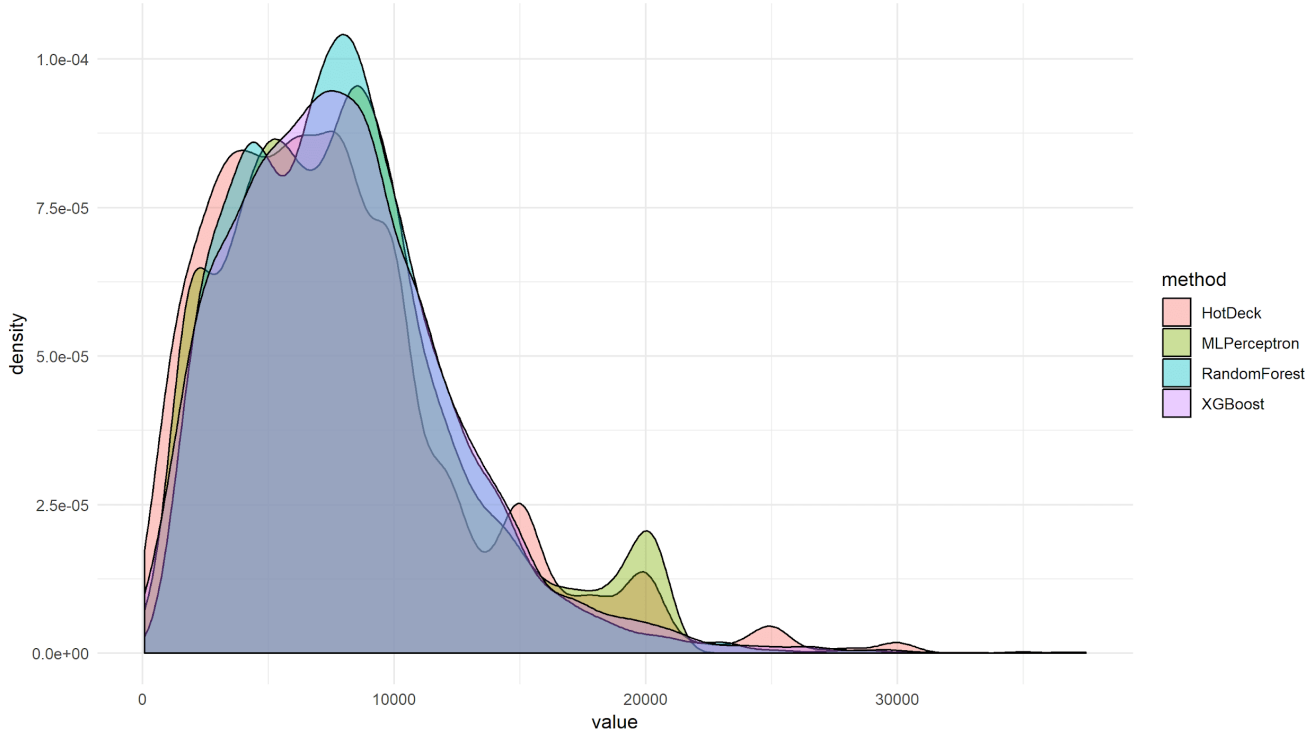
Se observa que los métodos de ensamble superan ampliamente la performance de hot deck. Particularmente, si se observa el RMSE (Rooted Mean Squared Error), Random Forest logra un capacidad de predicción del 52%, XGBoost, del 45% en RMSE y MLP del 38%. Valores similares se obtienen al analizar el MAE (Mean Absolute Error). Estos valores 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 DGEyC de la CABA con los mismos fines en la EAH.

Ahora bien, ¿qué impactos 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 métodos? Para realizar una primera aproximación 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.

Comparación de distribuciones de datos imputados

Gráfico 2. Density plot de la variable ingresos laborales (p21) por método de imputación -casos imputados-



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

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.

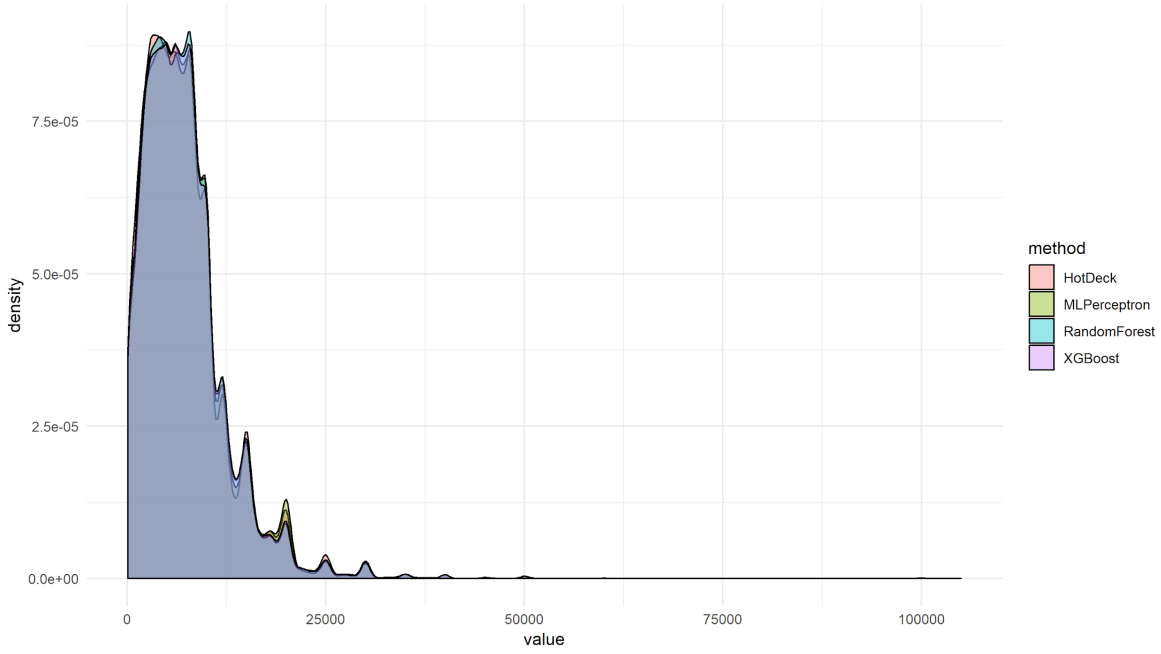
Tabla 4. Estadísticos descriptivos de la variable ingresos laborales (p21) según método de imputación -casos imputados-

Algoritmo	Min	Q1	Q2	Media	Q3	Max	CV
Hot Deck	$200.0	$4000.0	$7000.0	$7729.0	$10000.0	$35000.0	66.4%
MLPerceptron	$111.0	$4623.0	$7619.0	$8049.0	$10289.0	$20678.0	57.2%
Random Forest	$679.0	$4680.0	$7588.0	$7879.0	$10208.0	$35579.0	52.4%
XGBoost	$93.0	$4706.0	$7472.0	$7939.0	$10424.0	$37492.0	56.2%

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

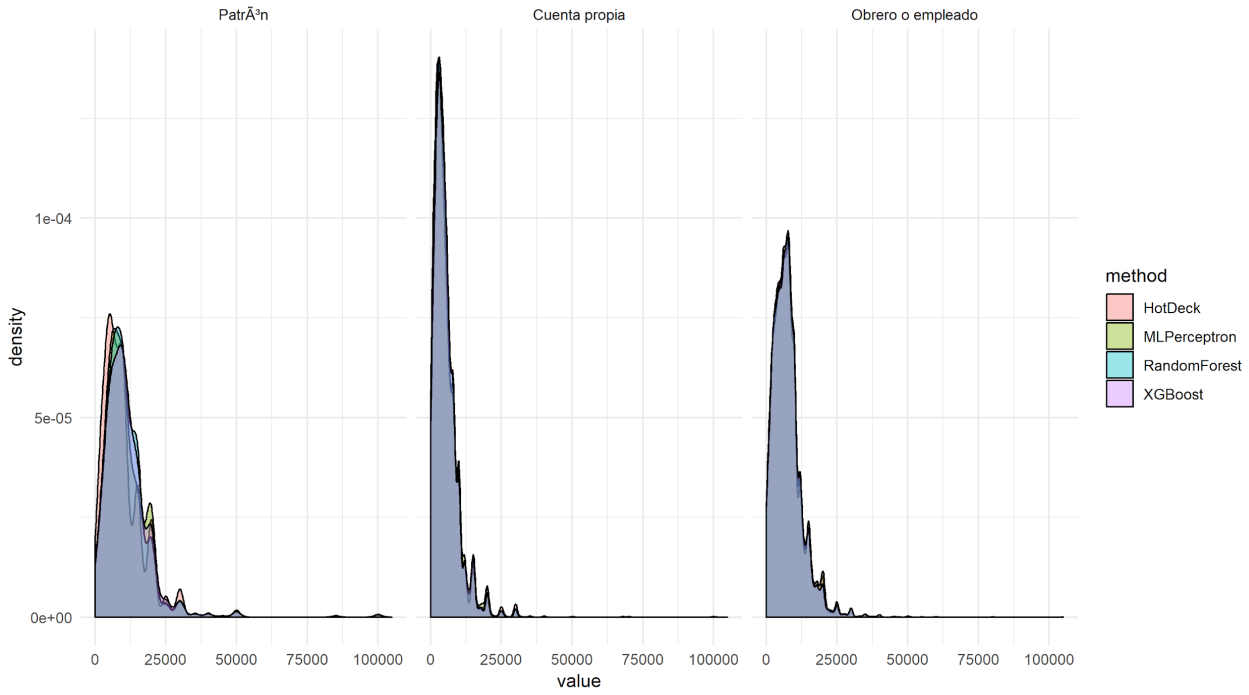
Los descriptivos anteriores confirman la imagen del gráfico: similitud en el centro de la distribución y divergencia en las colas. Particularmente relevante es notar que la dispersión es elevada en todos los métodos, pero es notablemente más elevada en el caso de Hot Deck.

Gráfico 3. Density plot de la variable ingresos laborales (p21) por método de imputación -casos imputados y completos-



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

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



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

Al comparar, ahora, la distribución total de la variable (es decir, casos imputados y completos) A su puede verse que no parece alterarse sustancialmente. Lo mismo sucede, si se analiza la distribución condicionada a las dos grandes categorías ocupacionales: asalariados y trabajadores independientes (patrones y trabajadores por cuenta propia).

Discusión

En el presente documento se presentaron los primeros 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 laborales 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), mejoras que van,

en términos relativos, desde el 30% hasta el 50%. 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. 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 próximas 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. ¿Qué sucede con los intervalos de confianza de los estimadores clásicos como el coeficiente de Gini o incluso con la estimación de la pobreza? ¿Qué efecto tiene cada uno de los métodos de imputación sobre los valores de estos indicadores?[8]

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 a 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 convnets o resnets).

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[9].

[8] Martín Montane (miembro del equipo) se encuentra trabajando en ambas líneas.
[9] María Giselle Galli, miembro del equipo, se encuentra desarrollado su tesis de maestría en esta dirección.

Referencias bibliográficas

Allison, P. (2002). *Missing Data*. Sage University Papers on Quantitative Applications in the Social Sciences . 07-136. California: Sage.

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

Breiman, L. (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". 2° 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.

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)". *14° 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. (2017). *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.

Hoerl, A. E. y Kennard, R. (1970). "Ridge regression: Biased estimation for nonorthogonal problems". *Technometrics*, 12. 55-67.

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". *14° 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, INDEC, Buenos Aires.

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. X Congreso de Nacional de Estudios del Trabajo, (ASET), Buenos Aires.

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

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.

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). Revista Estudios del Trabajo, 18. 93-110.

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

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society, 58. 267–288. Zhou, Z. (2012). Ensamble Methods. Foundations and Algorithms. Florida: Chapman & Hall/CRC