EL COLEGIO DE MÉXICO
CENTRO DE ESTUDIOS ECONÓMICOS

MAESTRÍA EN ECONOMÍA

TRABAJO DE INVESTIGACIÓN PARA OBTENER EL GRADO DE MAESTRO EN ECONOMÍA

PREDICCIÓN DE LA POBREZA LABORAL EN MÉXICO MEDIANTE ÍNDICES DE CORTO PLAZO BASADOS EN ALGORITMOS DE MACHINE LEARNING

RATZANYEL DANIEL RINCÓN VARGAS

PROMOCIÓN 2016-2018

ASESOR:

DR. ENEAS ARTURO CALDIÑO

JUNIO 2018

Agradecimientos especiales a mis padres y a mi compañera de vida:

Ana Bertha Vargas Hernández
Daniel Rincón Arroyo
María Belén Chávez y Palma

Así como a todos los profesores, amigos y personas que siempre me apoyaron durante todo este proceso, sin ustedes esto no hubiera sido posible.

¡Muchas gracias a todos!

Resumen

La medición de la pobreza es una tarea igual de importante que el combatirla. Es por ello que el presente trabajo explora la posibilidad de utilizar algoritmos de Machine Learning para predecir la pobreza multidimensional a nivel nacional en el corto plazo, con la finalidad de ayudar a las autoridades a focalizar de mejor manera los recursos y las políticas de desarrollo social.

De los tres algoritmos analizados desde 2010 hasta 2017, los Bosques Aleatorios y las células de Máquinas de Vectores de Soporte con kernel Gaussiano son los que presentan las tasas de error más bajas a lo largo del tiempo, mientras que el mejor estimador de corto plazo de la pobreza oficial resultó ser el indicador compuesto por el Índice de la Tendencia Laboral de la Pobreza y los Bosques Aleatorios. Por otro lado, la evaluación conjunta de los algoritmos mostró que los Bosques Aleatorios son el modelo de clasificación de la pobreza más robusto y consistente a través del tiempo.

Índice

1
Introducción

La medición de la pobreza siempre ha sido de gran importancia tanto para los gobiernos de todo el mundo como para organizaciones internacional como el Banco Mundial. El identificarla de manera adecuada y saber medirla es igual de importante que el combatirla, principalmente porque una buena identificación de las zonas más necesitadas puede llevar a focalizar de mejor manera los programas sociales y demás políticas del gobierno para reducirla.

Desde el año 2004 en México, la Ley General de Desarrollo Social (LGDS) estableció la creación del Consejo Nacional de Evaluación de la Política de Desarrollo Social (CONEVAL), el cual está encargado de la medición de la pobreza y la evaluación de la política de desarrollo social del país. Este organismo gubernamental presenta las estimaciones oficiales de la pobreza cada dos años a nivel estatal y cada cinco a nivel municipal, tal y como lo indica la LGDS. Su principal fuente de información es la Encuesta Nacional de Ingresos y Gastos de los Hogares (ENIGH) junto con su Módulo de Condiciones Socioeconómicas (MCS), que se levantan por el Instituto Nacional de Estadística y

Geografía (INEGI) con una periodicidad de dos años.

Además de la medición de la pobreza, el artículo 36 de la LGDS [2] puntualiza que para su identificación, el CONEVAL debe tomar en cuenta los siguientes nueve indicadores:

I. Ingreso corriente per cápita.

II. Rezago educativo promedio en el hogar.

III. Acceso a los servicios de salud.

IV. Acceso a la seguridad social.

V. Calidad y espacios de la vivienda.

VI. Acceso a los servicios básicos en la vivienda.

VII. Acceso a la alimentación nutritiva y de calidad.

VIII. Grado de cohesión social.

IX. Grado de Accesibilidad a carretera pavimentada.

Así pues, como parte integral de la medición de la pobreza y desde una perspectiva opuesta a la visión tradicionalista unidimensional,[1] el CONEVAL desarrolló los criterios oficiales para determinarla mediante la Metodología para la medición multidimensional de la pobreza en México [9]. En esta metodología se especifican los lineamientos y criterios a seguir para la identificación de la pobreza de un individuo de acuerdo a su situación de bienestar económico, sus derechos sociales y el contexto territorial.

[1]Se dice de carácter unidimensional ya que únicamente se utiliza al ingreso como una aproximación del bienestar económico de la población [9].

El análisis del primer indicador de la LGDS se realiza en el espacio de bienestar, en donde se toma en cuenta la cantidad mínima de ingresos monetarios requerida para satisfacer las necesidades básicas de las personas. Mientras que en los espacios de derechos sociales y contexto territorial se analizan los indicadores del II al VII. Por consiguiente, de acuerdo con la metodología oficial, la pobreza se puede definir de la siguiente manera:

> Una persona se encuentra en situación de pobreza multidimensional cuando no tiene garantizado el ejercicio de al menos uno de sus derechos para el desarrollo social, y si sus ingresos son insuficientes para adquirir los bienes y servicios que requiere para satisfacer sus necesidades. [9].

En pocas palabras, se dice que un individuo es pobre si su ingreso es inferior a la linea de bienestar[2] y padece al menos un tipo de las seis carencias sociales.[3]

Todo este marco teórico desarrollado por el CONEVAL ha servido para el mejoramiento de la medición de la pobreza en México en los últimos años. Sin embargo, la frecuencia tan espaciada de sus resultados (cada dos años) presenta una gran desventaja, ya que la mayoría de los programas sociales y políticas orientadas a combatir la pobreza presuponen resultados de corto-mediano plazo, e incluso requieren de

[2]La linea de bienestar equivale al valor total de la canasta alimentaria y de la canasta no alimentaria por persona al mes (CONEVAL).

[3]Carencia por rezago educativo, de acceso a los servicios de salud, de acceso a la seguridad social, por la calidad y espacios de la vivienda, por servicios básicos en la vivienda, y de acceso a la alimentación. [9]

un seguimiento mucho más periódico y continuo.

Dicha problemática no es causada por el tiempo de procesamiento de la información por parte del CONEVAL, sino más bien se encuentra directamente ligada con la periodicidad de la fuente de información primal (la ENIGH y su MCS). Además, de acuerdo a lo reportado por el CONEVAL en [8], contar con una fuente de información como la ENIGH para realizar estimaciones de pobreza, con una periodicidad más corta resultaría sumamente costoso, debido a la extensa información que se obtiene de los hogares.

Es por ello que en 2010 el CONEVAL desarrolló el Índice de la Tendencia Laboral de la Pobreza (ITLP), un indicador alternativo de corto plazo el cual permite contar con información trimestral para estimar el estado de la pobreza en el país. Éste indicador tiene como fuente principal de información a la Encuesta Nacional de Ocupación y Empleo (ENOE), publicada trimestralmente desde el año 2005.

Cabe señalar que la metodología oficial para identificar la pobreza no se puede implementar en la ENOE, pues no presenta la misma estructura en sus cuestionarios que la ENIGH. Sin embargo,

> al menos en la dimensión de bienestar económico,..., el comportamiento de los ingresos laborales comparados con la línea de bienestar mínimo en cada trimestre puede proporcionar información útil sobre la evolución del componente de bienestar económico de la medición de la pobreza. [8].

Por lo que en la construcción del ITLP, una persona se identifica

como pobre laboralmente si su ingreso laboral mensual es inferior al promedio trimestral de los valores de la línea de bienestar mínimo.

A pesar de que el ITLP esta íntimamente ligado con la componente de bienestar económico de la metodología oficial de la medición de la pobreza, es importante mencionar que el CONEVAL deja muy claro que las cifras estimadas por el ITLP no representan una medición oficial de la pobreza en el país, de modo que su principal objetivo es informar de manera oportuna a los hacedores de política pública las tendencias en el corto plazo de indicadores altamente correlacionados con los cambios en la pobreza medida a través de los ingresos. En otras palabras, muestra la proporción de personas que no pueden adquirir la canasta alimentaria con el ingreso de su trabajo.

Pero, ¿qué tan bien estima las cifras oficiales el ITLP?.
Las cifras más recientes de la pobreza oficial se dieron a conocer el 30 de agosto de 2017 mediante el comunicado de prensa N. 09 del CONEVAL [10], mientras que las estimaciones del ITLP se presentaron en el comunicado de prensa N. 02 [11] el 14 de febrero de 2018.

La Figura 1.1 muestra las cifras oficiales de la pobreza a nivel nacional reportadas por el CONEVAL desde 2010 hasta 2016, así como los valores estimados de la pobreza laboral por el ITLP de cada trimestre desde 2010 hasta el cuarto trimestre de 2017. Claramente se puede observar que las estimaciones realizadas por el ITLP subestiman las cifras oficiales de la pobreza a nivel nacional. Más precisamente, según

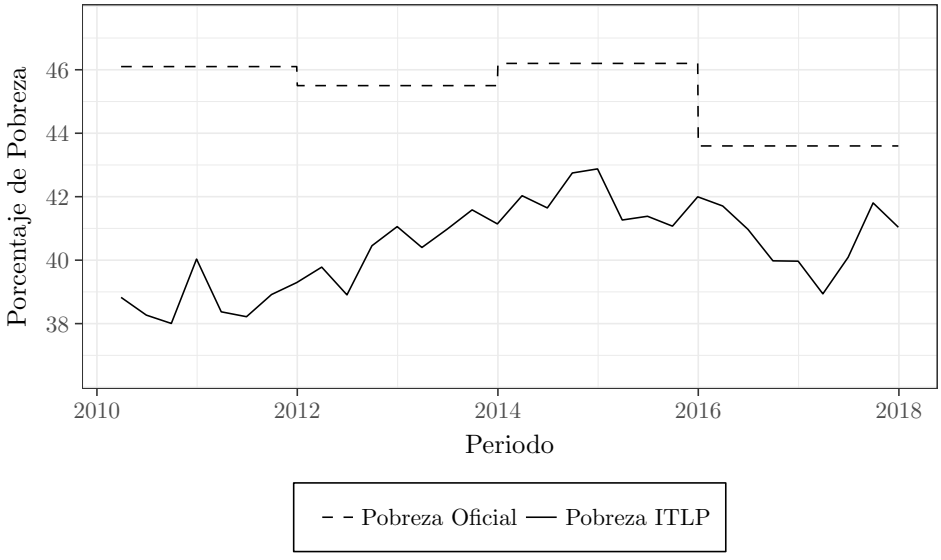

Figura 1.1: Pobreza laboral a nivel nacional estimada por el ITLP vs Cifras Oficiales de Pobreza, desde primer trimestre de 2010 hasta cuarto trimestre de 2017.

los números oficiales de CONEVAL, en el año de 2010 el 46.1 % de la población se encontraba en estado de pobreza, mientras que las estimaciones trimestrales de corto plazo realizadas por el ITLP oscilan entre el 38.8 %; i. e., una diferencia media de 7.3 % de la población total aproximadamente, o equivalentemente, 8.4 millones de personas[4] que no son catalogadas como pobres por el ITLP, pero que si lo son de acuerdo a la metodología oficial.

Por otro lado, realizando un análisis visual de la tendencia, se puede inferir que el ITLP refleja de manera adecuada la tendencia de

[4]Tomando en cuenta la población total nacional reportada por la ENIGH 2010.

la pobreza oficial en el país, principalmente a partir de 2014 donde se estabiliza para posteriormente decrecer en 2016.

La Figura 1.1 es la principal motivación de este trabajo, ya que plantea un campo fértil para la investigación de métodos de clasificación innovadores, con la finalidad de mejorar las estimaciones de corto plazo de la pobreza en México.

Es aquí donde entran los algoritmos de clasificación de Aprendizaje Automático, o *Machine Learning* por su nombre en inglés. A grandes rasgos, éstos métodos estadísticos tratan de encontrar patrones generalizables de los datos basándose en información preestablecida, con la finalidad de realizar predicciones de alguna variable de interés.

Dentro de la literatura existen un sin fin de aplicaciones de Machine Learning en diversas áreas del conocimiento. En el área de la salud por ejemplo, una pregunta natural sería si estos algoritmos de clasificación podrían ayudar a determinar si un paciente con diversas caracteristicas y sintomas podría ser suceptible o no a padecer de cáncer [12]. O si estos métodos estadísticos pueden ser capaces de realizar reconocimiento facial en imagenes digitales [14]. E incluso, llegar a pensar si estas herramientas predictivas harían alguna diferencia en el sistema acusatorio penal a la hora en la que el juez tome su decisión [19]. En todas ellas, se ha mostrado un buen desempeño del poder predictivo de las técnicas de Machine Learning , e incluso ha sobrepasado la precisión de los enfoques tradicionales.

El objetivo del presente trabajo es evaluar y comparar el desempeño de diversos algoritmos de Machine Learning tomando como información preestablecida a la base de datos de la ENIGH y su MCS (desde 2010 hasta 2016), para realizar estimaciones de la pobreza en la base de la ENOE. En otras palabras, identificar los patrones de los datos en la ENIGH y su MCS, para implementarlos en predicciones de la pobreza en la ENOE.

Debido a que el auge de las técnicas de Machine Learning es relativamente nuevo, no hay una basta literatura de la aplicación de los algoritmos en la Economía y mucho menos en la predicción de la pobreza en México. Pioneros en éste campo de investigación, Thomas Sohnesen y Niels Stender [24], utilizaron datos de seis paises diferentes[5] para comparar el desempeño de los Bosques Aleatorios (*Random Forest*) con el método de Imputación Múltiple, y encontraron que dentro de las estimaciones del mismo año el algoritmo de Machine Learning (Random Forest) era más preciso, sugiriendo así que este método podría contribuir a mejores predicciones de pobreza. Por otro lado, Linden McBride y Austin Nichols [22], presentaron evidencia de que la implementación de los algoritmos de Machine Learning en el desarrollo de los *Proxy Means Test* puede mejorar sustancialmente su poder predictivo fuera de la muestra.

Como dato adicional, a principios de 2018 el El Banco Mundial

[5]Albania, Etiopía, Malawi, Ruanda, Tanzania, y Uganda.

junto con la plataforma virtual Driven Data lanzaron una convocatoria a nivel mundial llamada *Pover-T Tests: Predicting Poverty*,[6] la cual básicamente invitaba a las personas a construir modelos para predecir el estado de pobreza de los hogares en tres países diferentes utilizando algoritmos de Machine Learning, con un total de quince mil dólares en premios. Reafirmando así que los trabajos en ésta área de investigación aún están en desarrollo.

El presente documento esta organizado de la siguiente manera. En el Capítulo 2 se presenta una breve descripción de los tres algoritmos de clasificación de Machine Learning utilizados: Análisis Discriminante Lineal, Bosques Aleatorios, y Máquinas de Vectores de Soporte; así como sus paquetes de estimación en el software estadístico *R*. En el Capítulo 3 se describen las bases de datos de entrenamiento (MCS-ENIGH) y de prueba (ENOE), junto con las variables que se homologaron entre ellas. En el Capítulo 4 se exponen los resultados obtenidos en la predicción de la pobreza por parte de los métodos descritos en el Capítulo 2, así como su comparación con las estimaciones hechas por el ITLP. Finalmente, en el Capítulo 5, se desarrollan las conclusiones del trabajo y se dan algunas recomendaciones para investigaciones futuras con el fin de mejorar las estimaciones de corto plazo de la pobreza en México.

[6]Disponible en: https://www.drivendata.org/competitions/50/worldbank-poverty-prediction/page/99/.

2

Algoritmos de Clasificación de Machine Learning

El problema de clasificación de la pobreza cae dentro de la rama del *aprendizaje supervisado* de Machine Learning. Este enfoque busca ajustar un modelo que relacione la variable objetivo Y a los predictores $X_1, ..., X_p$ dentro de una base de entrenamiento etiquetada, con la finalidad de predecir con precisión la respuesta para futuras observaciones y comprender mejor su relación. A diferencia de los métodos estándar utilizados en los que el objetivo principal es realizar una buena estimación de algún parámetro $\boldsymbol{\beta}$ que relacione a Y y $X_1, ..., X_p$, los algoritmos de clasificación de Machine Learning se enfocan más en la estimación de $\hat{Y}$.

En este capítulo se presentan tres de los algoritmos más utilizados en la literatura y que se pueden encontrar en cualquier libro de Machine Learning. Cada uno de ellos se desarrolla de forma breve incluyendo sus puntos más importantes: modelo matemático, interpretabilidad, parámetros de ajuste, procesos de validación, entre otros.

2.1. Análisis Discriminante Lineal

El modelo estadístico de Análisis Discriminante Lineal (ADL) fue propuesto por Fisher en 1936 para resolver el problema de predicción de variables cualitativas mediante variables independientes continuas, y es considerado como uno de los modelos clásicos de clasificación junto con la Regresión Logística.

Supóngase que se tiene una muestra de observaciones etiquetada $\{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\}$, donde $\boldsymbol{x}_i^T = (x_{i1}, ..., x_{ip})$ es el vector de características continuo del i-ésimo individuo, y cuya variable objetivo Y tiene dos clases posibles,[7] por ejemplo $y_i = 1$ si el i-ésimo individuo es pobre y $y_i = 0$ en caso contrario. La idea central del modelo es asignar la observación $\boldsymbol{x}_i$ a la clase k-ésima si su probabilidad *posterior* es la máxima, es decir, si

$$P(Y = k|\boldsymbol{X} = \boldsymbol{x}_i) \geq P(Y = k'|\boldsymbol{X} = \boldsymbol{x}_i),\ \forall k' \in \{1, 2\} \tag{2.1}$$

El clasificador ADL utiliza como valor predeterminado un umbral del 50 % para decidir a que clase asignar a las observaciones. No obstante, si por ejemplo lo que se busca es hacer más o menos sensible la clasificación de la clase k, se puede cambiar el valor de este parámetro de ajuste $\alpha \in (0, 1)$, de tal manera que la observación $\boldsymbol{x}_i$ se asigne a esta clase si $P(Y = k|\boldsymbol{X} = \boldsymbol{x}_i) > \alpha$.

[7] Tibshirani R. , James G., Witten D., y Hastie T. desarrollan el modelo de manera más general y detallada con K clases diferentes en [17].

Por otra parte, el supuesto clave de ADL es que las realizaciones de cada clase vienen de un vector aleatorio distribuido normal multivariado con una media especifica para cada clase, pero con una matriz de varianzas y covarianzas común entre clases. Más formalmente, las características observadas $\boldsymbol{x}_i$ para las cuales $y_i = k$ para alguna clase $k \in \{1, 2\}$, se asumen como realizaciones de un vector aleatorio

$$\boldsymbol{X}^k = (X_1^k, ..., X_p^k) \sim N_p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}) \tag{2.2}$$

Bajo estos supuestos, y con un poco de teoría de probabilidad y álgebra, se puede mostrar que la condición (2.1) es equivalente a asignar la observación $\boldsymbol{x}_i$ a la clase k-ésima si su *función discriminante* $\delta_k(\boldsymbol{x}_i)$ domina a todas las demás, es decir, si

$$\delta_k(\boldsymbol{x}_i) := \boldsymbol{x}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k - \frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \log \pi_k \geq \delta_{k'}(\boldsymbol{x}_i), \ \forall k' \in \{1, 2\}$$

donde π_k denota la probabilidad *previa* de que una observación pertenezca a la clase k. Nótese que los parámetros poblaciones involucrados en la definición de la función discriminante $\delta_k(\cdot)$ no son conocidos a *priori*, por lo que el enfoque de ADL los estima de la siguiente manera:

$$\begin{aligned} \hat{\boldsymbol{\mu}}_k &= \frac{1}{n_k} \sum_{i:y_i=k} \boldsymbol{x}_i \\ \hat{\sum} &= \frac{1}{n-2} \sum_{k=1}^{2} \sum_{i:y_i=k} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k)^T \\ \hat{\pi}_k &= \frac{n_k}{n} \end{aligned} \tag{2.3}$$

Con n_k igual al numero de observaciones de la k-ésima clase. Así pues, sustituyendo las estimaciones de (2.3) en la función discriminante se obtiene que

$$\hat{\delta}_k(\boldsymbol{x}) = \boldsymbol{x}^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_k - \frac{1}{2} \hat{\boldsymbol{\mu}}_k^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}_k + \log \hat{\pi}_k \tag{2.4}$$

Por lo tanto, el clasificador ADL asigna la observación $\boldsymbol{x}_i$ a la clase k para la cual $\hat{\delta}_k(\boldsymbol{x}_i)$ es máximo. Obsérvese que la función discriminante $\hat{\delta}_k(\boldsymbol{x})$ depende linealmente de $\boldsymbol{x}$; es decir, solo depende de $\boldsymbol{x}$ a través de una combinación lineal de sus componentes.

El proceso de validación del método ADL, consiste en la estimación de la tasa de error que tendría el modelo fuera de la base de entrenamiento. *10-fold* es uno de los enfoques de validación cruzada sugeridos por [17], el cual consiste en dividir aleatoriamente el conjunto total de observaciones de la base de entrenamiento en diez grupos disjuntos de tamaño aproximadamente igual $(n_1, ..., n_{10})$. El primer grupo se toma como un conjunto de validación, y el modelo se ajusta a los nueve grupos restantes. Luego, dentro del grupo que se dejo fuera, se calcula la tasa de error[8] $ER_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} I_{y_i \neq \hat{y}_i}$ de esta primera iteración. El procedimiento se repite nueve veces más, y en cada nueva iteración un grupo diferente a todos los anteriores es tomado como el grupo de validación. Finalmente, se habrán estimado diez tasas de

[8] $I_{y_i \neq \hat{y}_i}$ denota la función indicadora de $y_i \neq \hat{y}_i$.

error $ER_1, ..., ER_{10}$, por lo que la tasa de error estimada por 10-fold es el promedio $ER_{10fold} = \frac{1}{10}\sum_{i=1}^{10} ER_i$.

En el software estadístico *R*, ADL se encuentra dentro de la librería *MASS*, y se puede utilizar con la función *lda()*. Esta función da como valor de retorno las estimaciones de los coeficientes que multiplican a $\boldsymbol{x}$ en la función discriminante (2.4), las estimaciones de las probabilidades $\hat{\pi}_k$, y de los vectores $\hat{\boldsymbol{\mu}}_k$.

2.2. Bosques Aleatorios

Análisis Discriminante Lineal tiene un buen desempeño cuando los datos tienen fronteras de decisión lineales, sin embargo, únicamente se puede implementar cuando los predictores son continuos. Un método alternativo de clasificación que no presenta dichas limitaciones son los Bosques Aleatorios, o *Random Forests* por su nombre en inglés. Este método fue desarrollado por Leo Breiman [1] en 2001, aunque ya se tenían formulaciones iniciales por Tin Kam Ho en 1995. Para entender cómo funciona un Bosque Aleatorio (BA), primero se debe entender cómo funcionan los árboles de decisión (en particular los árboles de clasificación), ya que el clasificador de un BA esta compuesto por una colección de árboles de clasificación.

2.2.1. Árboles de Clasificación

Supóngase al igual que en ADL, que se tiene una muestra de observaciones etiquetada $\{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\}$, donde $\boldsymbol{x}_i^T = (x_{i1}, ..., x_{ip})$ es el vector de características (no necesariamente continuo) del i-ésimo individuo, y cuya variable objetivo Y tiene dos clases posibles. De acuerdo con [17], el proceso general de construcción de un *árbol de clasificación* consta de dos partes:

1. Dividir el espacio de los predictores $X_1, ..., X_p$ adecuadamente en J regiones disjuntas $R_1, ..., R_J$.

2. Asignar a las observaciones a la clase más frecuente dentro de la región R_j en la que pertenezcan.

Para construir las regiones $R_1, ..., R_J$ se utiliza una técnica denominada *división binaria recursiva*, en donde inicialmente se consideran todas las observaciones de la base de entrenamiento como una sola región. Luego, tomando un predictor X_j dado y un punto de quiebre[9] $s \in \mathbb{R}$, se construyen los *hiperplanos*

$$R_1(j, s) = \{\boldsymbol{X} | X_j \leq s\} \text{ y } R_2(j, s) = \{\boldsymbol{X} | X_j > s\}$$

Y para cada región $R_m(j, s)$ se obtiene la clase más frecuente k_m y el número de observaciones n_m.

[9]Si la variable X_j es cualitativa, entonces s es un subconjunto de sus valores posibles, y los hiperplanos quedan definidos como $R_1(j, s) = \{\boldsymbol{X} | X_j \in s\}$ y $R_2(j, s) = \{\boldsymbol{X} | X_j \notin s\}$.

Lo anterior se puede realizar computacionalmente para cada predictor $X_1, ..., X_p$ y punto de corte s, y por ende, es factible determinar a la pareja (X_j, s) que mejor divida a la región en cuestión, es decir, aquella que resuelva:

$$\underset{j,s}{\text{Min}} \left\{ \frac{1}{n_1} \sum_{\boldsymbol{x}_i \in R_1(j,s)} I_{y_i \neq k_1} + \frac{1}{n_2} \sum_{\boldsymbol{x}_i \in R_2(j,s)} I_{y_i \neq k_2} \right\} \tag{2.5}$$

Obsérvese que el problema (2.5) no es más que la minimización de la tasa de error total debido a la partición R_1 y R_2. Una vez que se encuentra la pareja óptima para la división del espacio, se repite el proceso pero esta vez tomando únicamente una de las dos regiones previamente identificadas. Al final de la segunda iteración se tendrán tres regiones identificadas, y así sucesivamente. El proceso continúa hasta que ninguna región contenga más de un número mínimo de observaciones n_{min}, dando como resultado una partición del espacio de características de tamaño J. Por último, k_j sería la clase pronosticada por el árbol de clasificación para una nueva observación $\boldsymbol{x} \in R_j$.

Más formalmente, los árboles de clasificación se pueden escribir como:

$$T(\boldsymbol{x}; \theta) = \sum_{j=1}^{J} k_j I_{\boldsymbol{x} \in R_j}$$

Donde $\theta = \{R_j, k_j\}_{j=1}^{J}$ es el parámetro que guarda toda la información relevante de su proceso de construcción.

La Figura 2.1 muestra gráficamente un árbol de decisión $T(\boldsymbol{x}; \theta)$ en

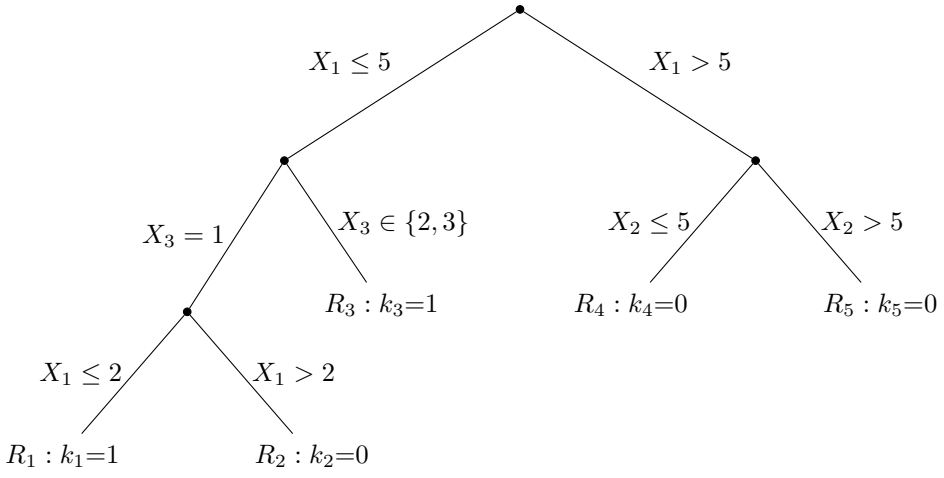

Figura 2.1: Árbol de clasificación $T(\boldsymbol{x}; \theta)$ con tres predictores, y una partición de cinco regiones.

donde el espacio de características es de dimensión tres, y la variable objetivo tiene dos clases posibles 0 y 1. Si se tuviera por ejemplo una nueva observación $\boldsymbol{x} = (1.3, 10, 1)$, entonces el árbol la clasificaría como $T(\boldsymbol{x}; \theta) = 1$, es decir, realizaría una estimación $\hat{y} = 1$ de su clase correspondiente.

Dos de las grandes ventajas de los árboles de clasificación son su gran interpretabilidad gráfica, pues son fáciles de explicar y entender, y su buen desempeño con fronteras de decisión altamente no lineales. Sin embargo, el método carece de robustez, pues un pequeño cambio en los datos de entrenamiento puede causar un gran cambio en el árbol de clasificación resultante (sufren de gran varianza).

2.2.2. Bosque Aleatorio

El problema de la varianza de los árboles de clasificación se puede solucionar mediante la adición de muchos de ellos como un solo predictor. Un Bosque Aleatorio utiliza esta idea para construir un modelo de predicción más poderoso y con poca varianza, es decir, en el cual se obtengan resultados similares si se aplica a bases de entrenamiento semejantes.

Sea $Z = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\}$ la base de entrenamiento del modelo. De acuerdo con [15], el algoritmo para la construcción del clasificador de un BA es el siguiente

1. Para $b = 1$ hasta B:
    a. Se extrae una muestra *bootstrap*[10] Z_b^* de tamaño n de la base de entrenamiento Z.
    b. Se ajusta un árbol de clasificación $T(\boldsymbol{x}; \theta_b)$ a la muestra Z_b^*, repitiendo recursivamente los siguientes pasos para cada nodo terminal del árbol, hasta que se alcance el tamaño mínimo de nodo n_{min}.
        i. Se seleccionan m variables al azar de los p predictores posibles $X_1, \ldots, X_p$.
        ii. De entre los m predictores, se elige la la variable óptima de división X_j y su correspondiente punto de quiebre s.
        iii. Se divide el nodo en dos nodos hijos.

[10] Una una muestra aleatoria con reemplazo de Z.

2. Como resultado se obtendrá un Bosque Aleatorio, i. e., conjunto de árboles de clasificación $\{T(\boldsymbol{x};\theta_b)\}_{b=1}^{B}$.

Cabe resaltar que este algoritmo reduce la varianza de los árboles de clasificación a costa de una disminución considerable de su interpretabilidad pues para una nueva observación $\boldsymbol{x}$, el clasificador de un BA predice la clase más votada por su conjunto de árboles, es decir,

$$\hat{y} = \text{Voto mayoritario } \{T(\boldsymbol{x};\theta_b)\}_{b=1}^{B}$$

Los parámetros de ajuste del modelo son tres principalmente, la cantidad de árboles en el Bosque Aleatorio (n_{trees}), el número de variables a tomar en cuenta en cada división (m), y el tamaño de observaciones mínimo de cada nodo terminal (n_{min}). Es importante mencionar que el parámetro m típicamente se toma igual a $\sqrt{p}$ para el problema de clasificación, con la finalidad de reducir la correlación entre los árboles.

El proceso de construcción de un BA permite realizar a la par una estimación de la tasa de error del modelo, pues en cada remuestreo bootstrap aproximadamente un tercio de las observaciones originales es dejado fuera. Al final del proceso, para cada observación $\boldsymbol{x}_i$ es posible realizar la predicción de su clase correspondiente utilizando los árboles en los cuales no se tomó en cuenta, y proceder con la regla del voto mayoritario para tener una única estimación $\hat{y}_i$. Así pues, la tasa de error estimada sería $ER = \frac{1}{n}\sum_{i=1}^{n} I_{y_i \neq \hat{y}_i}$.

Dentro de la librería *randomForest* de *R* se encuentra una función llamada *randomForest()*, la cual realiza toda la construcción del modelo en la base de entrenamiento, y da como resultado tanto el clasificador del BA como la estimación de la tasa de error fuera de la muestra.

2.3. Máquinas de Vectores de Soporte

Otra herramienta de aprendizaje supervisado para la clasificación binaria que se ha popularizado en los últimos años es el de las Máquinas de Vectores de Soporte,[11] o *Support Vector Machines* por su nombre en inglés. La idea central de este algoritmo consiste en producir fronteras de decisión no lineales mediante la construcción de fronteras lineales en una versión transformada y aumentada del espacio de características. Al igual que los Bosques Aleatorios, este método forma parte de los algoritmos de *caja negra* de Machine Learning, dado que solo se observan las entradas y las salidas pero no el proceso interno.

Las Máquinas de Vectores de Soporte (MVS) son una generalización del Clasificador de Vectores de Soporte (CVS), por lo que primero se presenta una breve descripción de él, y posteriormente se extiende la idea para ajustar fronteras de decisión no lineales y dar como

[11] Desarrollada en los noventas por Vladimir N. Vapnik.

resultado al clasificador de MVS.

2.3.1. El Clasificador de Vectores de Soporte

El Clasificador de Vectores de Soporte está basado en la separación natural que produce un hiperplano en el espacio de características $X_1, \ldots, X_p$. Considérese como en los métodos anteriores, una base de entrenamiento $\{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\}$, donde $y_i \in \{-1, 1\}$ es la variable binaria que determina a que clase pertenece el i-ésimo individuo. Defínase un *hiperplano* $H(\boldsymbol{\beta}, \beta_0) \subset \mathbb{R}^p$ como

$$H(\boldsymbol{\beta}, \beta_0) = \{\boldsymbol{x} \in \mathbb{R}^p \mid \boldsymbol{x}^T\boldsymbol{\beta} + \beta_0 = 0\}$$

Donde $\boldsymbol{\beta} \in \mathbb{R}^p$ es un vector unitario y $\beta_0 \in \mathbb{R}$ es el intercepto. Nótese que este hiperplano está totalmente caracterizado por los valores de $\boldsymbol{\beta}$ y β_0. Más aún, $H(\boldsymbol{\beta}, \beta_0)$ proporciona una forma natural de clasificación binaria en un espacio p-dimensional dada por

$$y(\boldsymbol{x}) = \begin{cases} -1, & \text{si } \boldsymbol{x}^T\boldsymbol{\beta} + \beta_0 < 0 \\ 1, & \text{si } \boldsymbol{x}^T\boldsymbol{\beta} + \beta_0 \geq 0 \end{cases} \tag{2.6}$$

Por ahora, supóngase que es posible encontrar un hiperplano $H(\boldsymbol{\beta}, \beta_0)$ que separe de manera perfecta a las observaciones en la base entrenamiento de acuerdo con sus respectivas clases; i. e., que los datos de entrenamiento sean *linealmente separables*, tal y como lo muestra la

Figura 2.2. Nótese que bajo este supuesto, existen una infinidad de hiperplanos que podrían ser usados para construir el clasificador (2.6), por lo que se utiliza el criterio del *hiperplano de margen máximo* para decidir cual es el mejor. Este criterio básicamente calcula la distancia perpendicular mínima (también conocida como *margen*) de las observaciones $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ a un cierto hiperplano de separación $H(\boldsymbol{\beta}, \beta_0)$, y se define al mejor como aquel que tenga la distancia mínima más grande, i. e., aquel que presente el margen máximo. Al clasificador (2.6) que resulta de considerar el hiperplano de margen máximo se le conoce como el *clasificador de margen máximo*. Una característica importante de este clasificador es que depende exclusivamente de aquellas observaciones que se encuentran más cercanas a él (llamados *vectores de soporte*).

Sin embargo, el clasificador de margen máximo está basado en el supuesto de la existencia de un hiperplano de separación perfecta, lo cual no siempre ocurre en la realidad. Para resolver este inconveniente, se recurre al Clasificador de Vectores de Soporte, el cual generaliza la idea del clasificador de margen máximo permitiendo que algunas de las observaciones se encuentren dentro del margen e incluso del lado equivocado del hiperplano de separación, añadiendo robustez al método.

Más precisamente, el hiperplano de separación $H(\boldsymbol{\beta}, \beta_0)$ escogido por el CVS es aquel que resuelve el problema de optimización primal

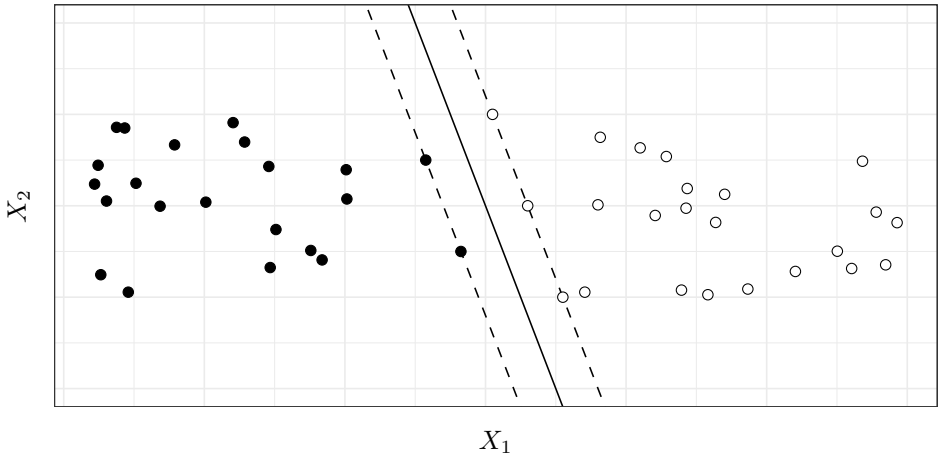

Figura 2.2: Hiperplano de margen máximo y sus vectores de soporte asociados, con datos linealmente separables en un espacio con dos predictores.

$$\begin{aligned} &\underset{\boldsymbol{\beta},\beta_0,\boldsymbol{\xi}}{\text{Min}} \left\{ \frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + C\sum_{i=1}^{n}\xi_i \right\} \\ &\text{S.a.} \quad y_i(\boldsymbol{x}_i^T\boldsymbol{\beta} + \beta_0) \geq 1 - \xi_i,\ \xi_i \geq 0,\ i = 1, ..., n. \end{aligned} \tag{2.7}$$

Donde $\boldsymbol{\xi}^T = (\xi_1, \ldots, \xi_n)$ es el vector de *variables de holgura* que permiten que las observaciones caigan dentro del lado incorrecto del margen, e incluso del lado incorrecto del hiperplano de separación, y C es un parámetro de afinación no negativo (mejor conocido como *parámetro de costo*) el cual se estima utilizando validación cruzada (10-fold). En este caso, los vectores de soporte son las observaciones que caen sobre el margen o en el lado incorrecto del mismo para su propia clase.

El problema descrito en (2.7) presenta el inconveniente de que $\boldsymbol{\beta}$

puede ser demasiado grande y dificultar los métodos computacionales para su resolución, por lo que un problema de optimización más simple de resolver es el correspondiente planteamiento dual.[12]

Así pues, resolviendo computacionalmente el problema dual de (2.7) se pueden obtener las estimaciones para los parámetros $\hat{\boldsymbol{\beta}}$ y $\hat{\beta}_0$, por lo que el CVS modelado por el algoritmo se encuentra dado por

$$\hat{y}(\boldsymbol{x}) = \begin{cases} -1, & \text{si } \boldsymbol{x}^T\hat{\boldsymbol{\beta}} + \hat{\beta}_0 < 0 \\ 1, & \text{si } \boldsymbol{x}^T\hat{\boldsymbol{\beta}} + \hat{\beta}_0 \geq 0 \end{cases} \tag{2.8}$$

2.3.2. Máquina de Vectores de Soporte

El Clasificador de Vectores de Soporte realiza un buen desempeño en la clasificación de datos que sean aproximadamente linealmente separables. Sin embargo, en la vida real existen muchas relaciones no lineales entre los predictores y su clase respectiva. Por ello, las Máquinas de Vectores de Soporte tratan de extender el alcance del CVS de tal forma que sea posible establecer fronteras de decisión no lineales en el espacio de características original, mediante fronteras de decisión lineales en un espacio de predictores transformado.

Más formalmente, se escoge una función *kernel*[13] $K : \mathbb{R}^p \times \mathbb{R}^p \rightarrow$

[12] Véase [15] para un desarrollo más detallado.

[13] Intuitivamente, el kernel es una función que cuantifica la similaridad entre observaciones.

Kernel	$K(\boldsymbol{x}, \boldsymbol{x}')$	Parámetros
Lineal	$\boldsymbol{x}^T\boldsymbol{x}'$	Ninguno
Polinomial de grado d	$(\gamma\boldsymbol{x}^T\boldsymbol{x}' + r)^d$	d, r, γ
Gaussiano	$\exp(-\gamma \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|^2)$	γ

Tabla 2.1: Funciones Kernel y sus parámetros de ajuste.

$\mathbb{R}_+$ para determinar la transformación $h = (h_1, ..., h_m) : \mathbb{R}^p \to \mathbb{R}_m$ que convertirá el espacio de características de dimensión p en uno de dimensión mayor (incluso infinito). Donde K y h se relacionan por la expresión

$$K(\boldsymbol{x}, \boldsymbol{x}') = h(\boldsymbol{x})^T h(\boldsymbol{x}'),\ \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^p \tag{2.9}$$

Nótese que no es necesario especificar la transformación h en primera instancia, pues se requiere únicamente información sobre la función kernel K para determinarla mediante la relación (2.9). La razón por la que se utilizan funciones kernel para determinar las funciones base y no cualquier transformación h es principalmente computacional, pues los cálculos para obtener las estimaciones de los parámetros se pueden volver muy laboriosos e incluso imposibles si el espacio de características aumentado es demasiado grande. La Tabla 2.1 muestra algunos de los kernels más utilizados dentro de la literatura.

Una vez determinadas las funciones base $h_j, j = 1, ..., m$ mediante la especificación del kernel, se construye el CVS utilizando como base de entrenamiento a las observaciones transformadas $\{(h(\boldsymbol{x}_1), y_1), ...,$

$(h(\boldsymbol{x}_n), y_n)\}$, dando como resultado al clasificador de vectores de soporte $\hat{y}_K$ asociado al kernel $K(\cdot,\cdot)$. Donde

$$\hat{y}_K(\boldsymbol{x}) = \begin{cases} -1, & \text{si } h(\boldsymbol{x})^T\hat{\boldsymbol{\beta}} + \hat{\beta}_0 < 0 \\ 1, & \text{si } h(\boldsymbol{x})^T\hat{\boldsymbol{\beta}} + \hat{\beta}_0 \geq 0 \end{cases} \tag{2.10}$$

Comúnmente, el clasificador (2.10) es llamado *Máquina de Vectores de Soporte*, por lo que se tendrá una maquina distinta por cada kernel utilizado. Obsérvese que la función de decisión $\hat{y}_K$ es muy difícil de interpretar (si no que imposible) por lo que se considera un método con un alto grado de complejidad.

En *R*, la librería *e1071* contiene la función *svm()*, la cual toma como argumentos principales el kernel deseado y los valores de sus parámetros asociados, así como el valor del parámetro de costo C. En general, se requiere probar varias combinaciones de funciones kernels y sus respectivos parámetros para obtener el mejor modelo de MVS.

3
Datos

El entrenamiento de los algoritmos descritos en el Capítulo 2 se realizó sobre datos obtenidos de la ENIGH y su Módulo de Condiciones Socioeconómicas, correspondientes a los años 2010, 2012, y 2014, mientras que para 2016 se utilizó el Modelo Estadístico 2016 para la continuidad del Módulo de Condiciones Socioeconómicas de la Encuesta Nacional de Ingresos y Gastos de los Hogares (MEC 2016 del MCS-ENIGH). Por otra parte, su aplicación se implementó en las observaciones de la ENOE correspondientes a todos los trimestres desde el año 2010 hasta el 2017. Sin embargo, estas encuestas estás diseñadas para objetivos distintos, y por lo tanto cuentan con muchas variables no compatibles entre ellas. La sección 3.1 presenta las variables homologables entre las dos encuestas que se utilizaron para la aplicación de los algoritmos de clasificación.

Por otro lado, dado que algunos de los algoritmos de Machine Learning son relativamente sensibles a los datos utilizados, la sección 3.2 describe similitudes y diferencias tanto de las tendencias como de las proporciones reportadas en las bases de entrenamiento y de prueba.

3.1. Variables Homologadas

Las variables homologadas entre la ENIGH-MCS y la ENOE se dividen en cuatro bloques principalmente. El primer bloque es el *sociodemográfico*, constituido por la entidad federativa, la variable indicadora de las localidades rurales, el sexo, la edad, y el tamaño del hogar. El segundo corresponde al bloque *económico*, en donde se encuentran la población económicamente activa, las horas de trabajo a la semana por parte del hogar, el ingreso laboral mensual del hogar,[14] y una variable indicadora en caso de que sea menor que la línea de bienestar mínimo correspondiente. El tercer bloque es el de la *educación*, formado por una variable indicadora de inasistencia a la escuela, el nivel educativo de los individuos, y el indicador de carencia por rezago educativo de la metodología oficial de la pobreza. Finalmente, el bloque de la *salud* consta de las variables indicadoras de adscripción al IMSS, al ISSSTE, y de no acceso a ningún servicio de salud.

La Tabla 3.1 resume los bloques de variables homologadas descritos anteriormente. Además, muestra las variables de la ENOE y de la ENIGH-MCS que se utilizaron para crearlas. Cabe que señalar que las variables de la ENOE son tal cual aparecen reportadas por el INEGI, mientras que las variables de la ENIGH-MCS, a excepción de *tamh* y *htrab*, son construidas tal y como lo hace CONEVAL en sus reportes

[14] Deflactado a precios de agosto del año correspondiente.

Nombre	Descripción	ENIGH-MCS	ENOE
ent	Entidad	ent	ent
rururb	Id. de localidades rurales	rururb	rururb
sexo	Sexo	sexo	sex
edad	Edad	edad	eda
tamh	Tamaño del hogar	creada	creada
pea	Población económicamente activa	pea	clase1/clase2
htrabh	Horas trabajadas en la semana en el hogar	htrab	hrsocup
ing_lab	Ingreso laboral mensual	ing_lab	p6b2/p6c
pob	Ingreso laboral menor a la lbm	ing_lab	p6b2/p6c
inas_esc	Inasistencia a la escuela	inas_esc	cs_p17
niv_ed	Nivel educativo	niv_ed	cs_p13_1
ic_rezedu	Indicador de carencia por rezago educativo	ic_rezedu	cs_p17/cs_p13_1 cs_p13_2/cs_p15
imss	Adscripción al IMSS	serv_sal	imssisssste
issste	Adscripción al ISSSTE	serv_sal	imssisssste
no_salud	Sin servicio de salud	serv_sal	imssisssste

Tabla 3.1: Variables Homologadas.

oficiales de la pobreza.[15]

A pesar de ser relativamente pocas las variables homologadas, es importante mencionar que se recopilaron variables que no solo describen el ingreso laboral de los hogares, sino también el rezago educativo, y parte de los accesos a los servicios de salud con los que cuentan los hogares. Es decir, los datos homologados toman en cuenta tres de los nueve rubros establecidos por la LGDS a la hora de determinar la pobreza multidimensional, ampliando así el abanico de características para realizar predicciones sobre la pobreza en el corto plazo.

Aunado a las variables homologadas dentro de la ENIGH-MCS, se encuentra la variable objetivo *pobreza*, construida a partir de los procedimientos de cálculo del CONEVAL de acuerdo con la metodología oficial de la pobreza. Esta variable etiqueta y diferencia de forma oficial a los individuos pobres de los no pobres, y dado que no es posible replicarla de manera determinista en la ENOE, los algoritmos de Machine Learning se centran en su predicción.

[15]Programas de cálculo disponibles en: https://www.coneval.org.mx/Medicion/MP/Paginas/Programas_BD_10_12_14_16.aspx

3.2. Base de Entrenamiento y Base de Prueba

Dentro del lenguaje utilizado en Machine Learning, se les conoce como base de entrenamiento y base de prueba a las bases de datos resultantes de dividir en dos partes el total de las observaciones etiquetadas, de tal manera que una mayor proporción se utilice para entrenar a los algoritmos, y posteriormente implementarlos en la base de prueba para obtener una estimación de su precisión. Sin embargo, en el presente trabajo se le denomina *base de entrenamiento* al total de las observaciones etiquetas, es decir, a todos los registros homologados de la ENIGH-MCS. Mientras que, se refiere a la *base de prueba* como todas aquellas observaciones homologadas provenientes de la ENOE.

Las tablas de medias y desviaciones estándar de las variables homologadas para cada base se presentan en el Anexo A.[16] Las Tablas A.1 y A.3 a A.6 muestran las estadísticas de las variables a nivel hogar de las bases de entrenamiento y prueba, respectivamente. Es fácil identificar que todas, a excepción del ingreso (y por ende la variable *pob* también), han permanecido relativamente estables a través del tiempo, y además presentan cifras comparables entre encuestas.

Con respecto al ingreso, la Figura 3.1 muestra la evolución del

[16]Las estimaciones de las estadísticas descriptivas se realizaron sin utilizar el *factor de expansión poblacional*, con la finalidad de comparar los datos crudos.

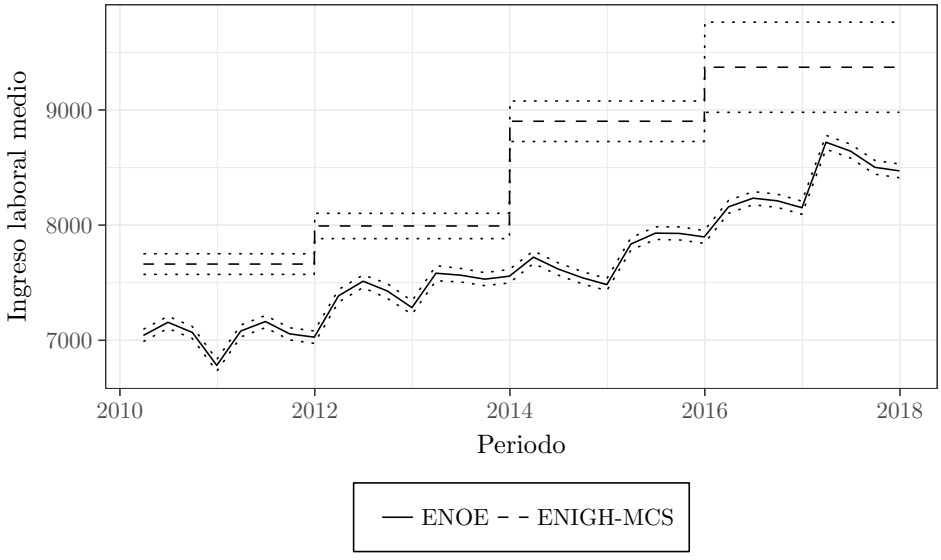

Figura 3.1: Ingreso laboral medio a nivel nacional de la base de Prueba y de Entrenamiento junto con sus intervalos de confianza al 95 %, desde el primer trimestre de 2010 hasta el cuarto trimestre de 2017.

ingreso laboral medio de los hogares mexicano a través del tiempo, reportados tanto en la base de entrenamiento como en la base de prueba. Si bien ambas tendencias coinciden al ser crecientes, sus intervalos de confianza hacen sospechar la existencia de diferencias estadísticamente significativas entre las medias. La Tabla A.11 presenta los resultados más importantes de la prueba *one-way* ANOVA, en donde se rechaza, a todos los niveles de significancia, la hipótesis nula de que en conjunto todas las medias sean estadísticamente iguales, dentro de su periodo correspondiente (2010 a 2011, 2012 a 2013, 2014 a 2015,

y 2016 a 2017). Cabe señalar que la gran dispersión del ingreso en la base de entrenamiento de 2016 es debida a que se utilizó el MEC 2016 del MCS-ENIGH, pues el cambio metodológico de la ENIGH 2016 realizado por el INEGI no permitía dar continuidad a la serie de pobreza.

Por otro lado, las Tablas A.2 y A.7 a A.10, exhiben las estadísticas descriptivas de algunas de las variables homologadas a nivel individual de las bases de entrenamiento y prueba, respectivamente. Al igual que en las estadísticas a nivel hogar, la mayoría de ellas no muestran grandes cambios a través del tiempo ni cambios relevantes de una encuesta a otra. Sin embargo, vale la pena mencionar que la variable de no atención médica tiene una tendencia a la baja en la base de entrenamiento, iniciando en 27 % de la población nacional en 2010 y cayendo abruptamente hasta un 14 % en 2016. Mientras que en la base de prueba se mantiene fluctuando a través del tiempo al rededor del 25 % .

Finalmente, la variable objetivo presenta una desviación estándar prácticamente constante a través del tiempo de 0.499 , y una media balanceada, en el sentido de que ninguna clase (pobre o no pobre) supera el 60 % de las observaciones totales, y permanece fluctuando muy cerca del 45 %. Cabe señalar que estas cifras crudas se convierten en las estimaciones oficiales de la pobreza graficadas en la Figura 1.1 cuando se aplica el factor de expansión poblacional.

4

Resultados y discusión

La implementación sobre la base de prueba de los algoritmos ya entrenados se dividió en distintos periodos según los años de la base de entrenamiento. Más precisamente, aquellos modelos generados con la base de entrenamiento de 2010 se utilizaron para predecir la pobreza en la base de prueba correspondiente a todos los trimestres de 2010 y 2011, los modelos entrenados en 2012 se emplearon para predecir la pobreza en todos los trimestres de la base de prueba de 2012 y 2013, y así sucesivamente.

Las secciones 4.1, 4.2, y 4.3 describen los resultados a nivel nacional de la estimación de la tasa de la pobreza realizada por los tres algoritmos presentados en el Capítulo 2, mientras que sus estimaciones precisas se muestran en las tablas del Anexo C. Por otro lado, la sección 4.4 presenta un análisis de robustez de los algoritmos implementados mediante una comparación de su calidad a través del tiempo, evaluada por las diferentes métricas descritas en el Anexo D.

4.1. Índice de la Pobreza de ADL

Dado que Análisis Discriminante Lineal solo puede ser utilizado con variables continuas, se omitieron todas las variables homologadas que fueran categóricas, restando únicamente las variables de edad, tamaño del hogar, horas trabajadas, y el ingreso laboral. Por otra parte, la literatura señala que el desempeño del algoritmo es mejor generalmente, si las variables se encuentran estandarizadas y si se incluyen algunas interacciones entre ellas, por lo que se procesaron de esta manera y se agregaron las interacciones entre el tamaño del hogar y el ingreso, y las horas trabajadas y el ingreso.[17]

Así pues, para 2010 por ejemplo, se obtuvieron las siguientes estimaciones de las funciones discriminantes[18]

$$\hat{\delta}_0(\boldsymbol{z}) = -0.004z_{edad} - 0.409z_{tamh} + 0.169z_{htrabh} + 0.437z_{ing} + 0.144(z_{tamh} \times z_{ing}) + 0.147(z_{htrabh} \times z_{ing}) - 0.799$$

$$\hat{\delta}_1(\boldsymbol{z}) = -0.034z_{edad} + 0.469z_{tamh} - 0.268z_{htrabh} - 0.694z_{ing} - 0.253(z_{tamh} \times z_{ing}) + 0.408(z_{htrabh} \times z_{ing}) - 0.627$$

Por lo tanto, ADL etiqueta al i-ésimo individuo como *Pobre* si:

$$\hat{\delta}_1(\boldsymbol{z}_i) \geq \hat{\delta}_0(\boldsymbol{z}_i)$$

[17]Las demás interacciones posibles no generaban una mejora significativa en la tasa de error del modelo.

[18]Tómese como $y = 0$ a la clase *No Pobre* y $y = 1$ a la clase *Pobre*.

	Funciones discriminantes							
	2010		2012		2014		2016	
	$\hat{\delta}_0$	$\hat{\delta}_1$	$\hat{\delta}_0$	$\hat{\delta}_1$	$\hat{\delta}_0$	$\hat{\delta}_1$	$\hat{\delta}_0$	$\hat{\delta}_1$
edad	-0.004	-0.034	-0.002	-0.029	-0.006	-0.010	0.002	-0.005
	(0.004)	(0.004)	(0.003)	(0.005)	(0.004)	(0.005)	(0.003)	(0.004)
tamh	-0.409	0.469	-0.393	0.440	-0.371	0.438	-0.319	0.475
	(0.004)	(0.004)	(0.004)	(0.005)	(0.005)	(0.005)	(0.004)	(0.005)
htrabh	0.169	-0.268	0.199	-0.299	0.219	-0.290	0.217	-0.278
	(0.005)	(0.007)	(0.006)	(0.008)	(0.007)	(0.009)	(0.004)	(0.010)
ing_lab	0.437	-0.694	0.382	-0.589	0.326	-0.591	0.759	-2.240
	(0.013)	(0.028)	(0.020)	(0.049)	(0.031)	(0.083)	(0.076)	(0.256)
tamh × *ing_lab*	0.144	-0.253	0.079	-0.254	0.179	-0.388	0.465	-1.336
	(0.010)	(0.017)	(0.015)	(0.035)	(0.037)	(0.068)	(0.067)	(0.238)
htrabh × *ing_lab*	0.147	0.408	0.158	0.367	0.035	0.397	-0.408	1.213
	(0.018)	(0.017)	(0.015)	(0.028)	(0.033)	(0.057)	(0.047)	(0.160)
constante	-0.799	-0.627	-0.784	-0.643	-0.710	-0.687	-0.575	-0.805
	(0.004)	(0.004)	(0.004)	(0.005)	(0.005)	(0.005)	(0.002)	(0.004)
Tasa de error	0.237		0.256		0.279		0.289	

Tabla 4.1: Estimaciones de los coeficientes de las variables continuas estandarizadas en las funciones discriminantes para los distintos periodos de la base de entrenamiento, y sus correspondientes errores estándar bootstrap.

Nótese que de acuerdo con las estimaciones obtenidas, la edad tiene una influencia casi nula a la hora de clasificar a los individuos, mientras que el ingreso laboral es la variable que más influye en la decisión. Además, el modelo refleja de manera correcta la intuición en las varia-

bles, pues la probabilidad de que se asigne como *Pobre* a un individuo aumenta si el tamaño del hogar es muy grande, o si el ingreso laboral mensual es muy pequeño, e incluso si las horas trabajadas a la semana son muy pocas.

La Tabla 4.1 presenta las estimaciones de las funciones discriminantes para todos los periodos de la base de entrenamiento. Cabe destacar que el efecto de la edad es casi nulo para todos lo años, mientras que el ingreso laboral es la variable que más pesa a la hora de la clasificación. De igual manera, los signos y magnitudes de los coeficientes correspondientes a 2012, 2014 y 2016 son similares a los obtenidos en 2010, por lo que la intuición que presenta ADL es consistente a través del tiempo.

Las tasas de error obtenidas del 10-fold de validación cruzada se presentan al final de la Tabla 4.1. Obsérvese que la tasa presenta una tendencia creciente a través de los años, aumentando de 23.7 % en 2010 hasta 28.9 % en 2016,[19] reflejando así un sesgo importante, principalmente en 2016, en las predicciones realizadas por el algoritmo. Las causas más evidentes de este sesgo son los pocos predictores utilizados, y la violación del supuesto de normalidad multivariada (2.2).[20]

[19]Una tasa de error del 28.9 % refleja que cualquier observación nueva es clasificada erróneamente por el algoritmo con una probabilidad de 0.289.

[20]Las Tablas B.1 y B.2 del Anexo B, presentan los resultados de las pruebas estadísticas de normalidad multivariable para ambas clases y todos los periodos de la base de entrenamiento. Nótese que en todos los casos se rechaza, a todos los niveles de significancia, la hipótesis nula de normalidad.

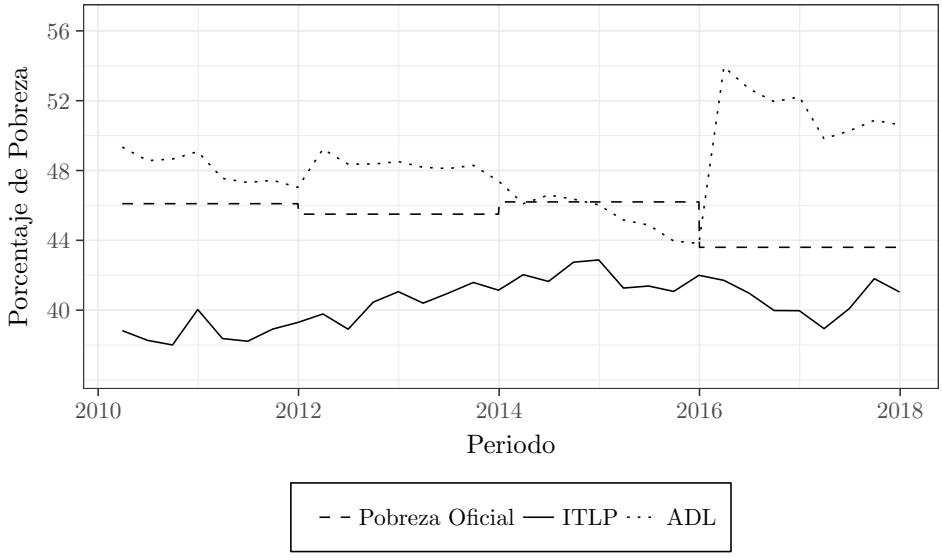

Figura 4.1: Pobreza laboral a nivel nacional estimada por el ITLP, Cifras Oficiales de Pobreza, y ADL, desde primer trimestre de 2010 hasta cuarto trimestre de 2017.

Pese a todo lo anterior, de 2010 a 2016 el indicador de la pobreza nacional de ADL realiza un mejor trabajo en las estimaciones de la pobreza oficial que el ITLP, tal y como lo muestra la Figura 4.1. No obstante, a partir de 2016 el indicador de ADL aumenta de manera abrupta por la gran dispersión del ingreso, poniendo de manifiesto su sensibilidad a los datos. Obsérvese que las diferencias de las proporciones de pobreza entre el ITLP y las predichas por ADL son estadísticamente significativas, sin embargo, las predicciones del algoritmo de Machine Learning reflejan una tendencia de la pobreza laboral distinta a la presentada por el ITLP. Además, las prediccio-

nes realizadas por ADL sobreestiman en la mayoría de los periodos a las cifras oficiales de la pobreza multidimensional, dificultando así su aceptación por parte de las autoridades.

4.2. Índice de la Pobreza de BA

A diferencia de Análisis Discriminante Lineal, los Bosques Aleatorios manejan variables categóricas sin ningún problema tal y como se mostró en el Capítulo 2. Por ende, para su desarrollo se utilizaron todas las características homologadas de la Tabla 3.1.

Por otro lado, recuérdese que el algoritmo de BA presenta tres parámetros de ajuste principales, a saber: la cantidad de árboles en el Bosque Aleatorio (n_{trees}), el número de variables a tomar en cuenta en cada división (m), y el tamaño de observaciones mínimo de cada nodo terminal (n_{min}). De acuerdo a la literatura, fijar $n_{trees} = 500$ y $n_{min} = 1$ resulta suficiente para un buen desempeño del algoritmo, por lo que la calibración del mismo se centra en escoger de manera adecuada el número m de variables a tomar en cuenta en cada división. Para ello se utilizó una *cuadrícula* o *grid* por su nombre en inglés, de valores distintos del parámetro m y se escogió el modelo que minimizara la tasa de error out-of-bag.[21] La Tabla 4.2 muestra los valores óptimos del parámetro m y las tasas de error obtenidas del

[21]Los valores de la cuadrícula utilizada fueron 2, 4, y 8.

Parámetro de ajuste de los BA				
	2010	2012	2014	2016
m	4	4	4	4
Tasa de error	0.147	0.143	0.146	0.166

Tabla 4.2: Número de predictores óptimo para cada *split* de los árboles de clasificación de los diferentes Bosques Aleatorios, y su correspondiente tasa de error *out-of-bag*.

proceso de calibración. No es sorpresa que el parámetro óptimo para todos los periodos haya resultado $m = 4$, pues como se mencionó en el Capítulo 2, $m = \sqrt{p} \approx 4$. Nótese que hay una mejora significativa en cada periodo de aproximadamente diez puntos porcentuales en la tasa de error de BA comparada con la obtenida en ADL, evidenciando que las fronteras de decisión son altamente no lineales y que la inclusión de los predictores categóricos genera un impacto positivo en el poder predictivo del algoritmo.

La Figura 4.2 muestra las predicciones de corto plazo de la tasa de pobreza en México realizadas por BA. Claramente las cifras obtenidas reflejan de manera correcta la tendencia del ITLP, sin embargo, subestiman aún más la pobreza oficial. Una posible justificación de esto es que las cifras de pobreza del ITLP toman en cuenta a todos los individuos cuyo ingreso per cápita es inferior a la línea de bienestar mínimo correspondiente, mientras que la metodología oficial de la

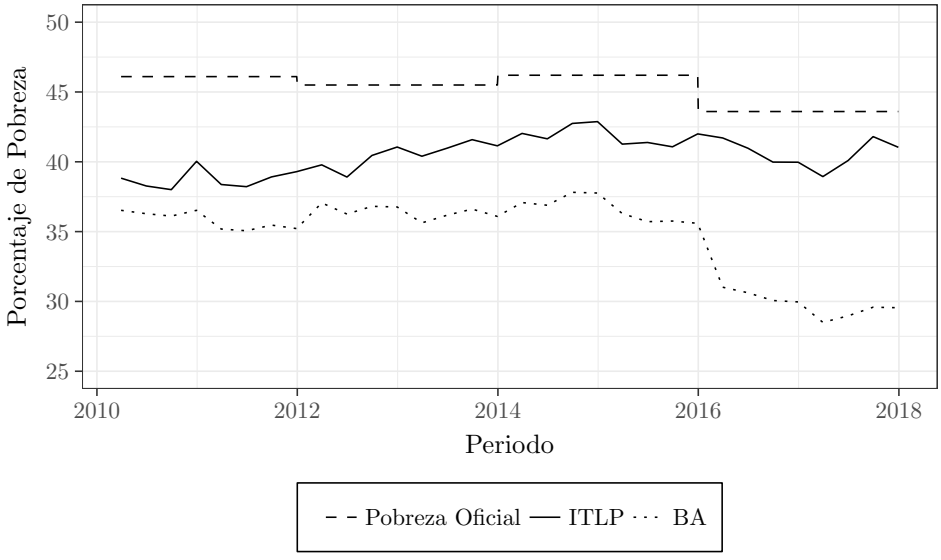

Figura 4.2: Pobreza laboral a nivel nacional estimada por el ITLP, Cifras Oficiales de Pobreza, y BA, desde primer trimestre de 2010 hasta cuarto trimestre de 2017.

pobreza añade la restricción de que deben contar además con al menos un tipo de carencia social. Es decir, idealmente el conjunto de los individuos pobres del ITLP debería contener a los individuos pobres identificados por BA en la base de prueba.

Un enfoque alternativo a los previamente planteados es el de complementar las herramientas proporcionadas por el ITLP y BA. Más precisamente, añadir a la base de individuos pobres del ITLP aquellos individuos que fueron clasificados como *Pobres* por el algoritmo de Machine Learning pero que son identificados como *No Pobres* en la base del ITLP. La intuición detrás de este indicador compuesto es

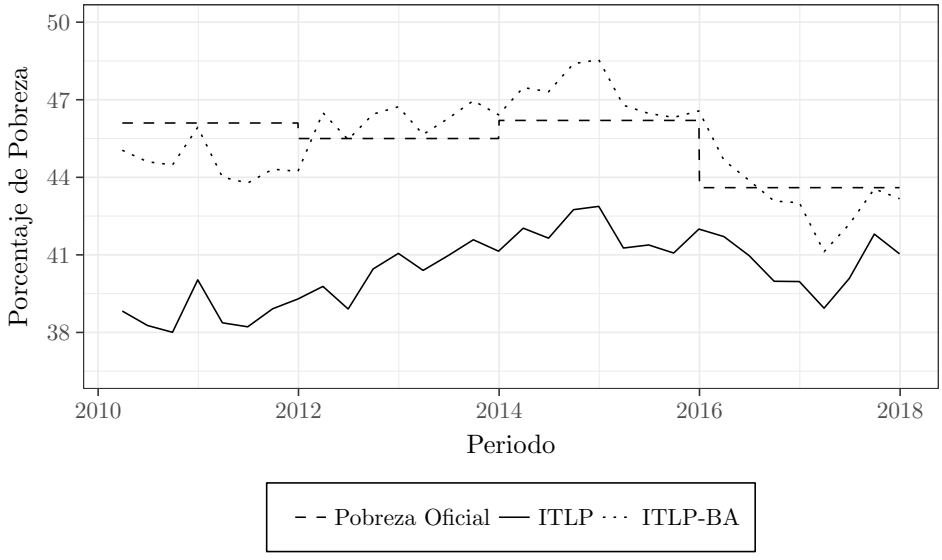

Figura 4.3: Pobreza laboral a nivel nacional estimada por el ITLP, Cifras Oficiales de Pobreza, e ITLP-BA, desde primer trimestre de 2010 hasta cuarto trimestre de 2017.

mejorar al ITLP con aquellos individuos inicialmente no pobres laboralmente, pero cuyas características presentan patrones de pobreza multidimensional identificados por el algoritmo de Machine Learning.

La Figura 4.3 muestra los resultados de esta agregación. Nótese que el indicador compuesto resultante ITLP-BA mejora de manera significativa las predicciones de la pobreza multidimensional en todos los periodos, y preserva la tendencia de su componente principal, el ITLP. Además, presenta mayor dinamismo con las cifras oficiales de la pobreza que los anteriores métodos, sobrepasando las problemáticas de la subestimación y sobreestimación permanente.

4.3. Índice de la Pobreza de MVS

Las Máquinas de Vectores de Soporte al igual que los Bosques Aleatorios, permiten trabajar con variables categóricas mediante la transformación de sus niveles en variables indicadoras, por lo que su desarrollo se efectuó también sobre todas las variables de la Tabla 3.1.

Es importante mencionar que los tiempos de entrenamiento de los algoritmos de MVS se incrementan demasiado rápido conforme al tamaño de la base de datos, por lo que se optó por implementar el enfoque alternativo propuesto por Steinwart y Thomann en [25]. Básicamente, la idea es particionar el espacio de características en un número de *células* pequeñas, con la finalidad de ajustar un clasificador de MVS en cada célula y así reducir considerablemente los tiempos de ejecución. Por lo tanto, una nueva observación $\boldsymbol{x}$ se clasificará según la predicción del clasificador de MVS de la célula a la que pertenezca. Es posible mostrar que la calidad de la solución bajo este enfoque es comparable con la de un único clasificador de SVM, al menos para dimensiones moderadas.

Por otra parte, en el Capítulo 2 se hizo énfasis en la importancia de la elección del kernel adecuado, por lo que siguiendo las recomendaciones de la literatura se utilizó el kernel Gaussiano debido a su buen rendimiento general y a su reducido número de parámetros (γ). Además, recuérdese que independientemente del kernel empleado, el

Parámetros de ajuste de MVS				
	2010	2012	2014	2016
N. Células	170	174	173	211
Tamaño medio	1,385	1,222	1,250	1,221
γ_{moda}	0.25	0.25	0.01	0.01
C_{moda}	10,000	10,000	10,000	10,000
Vect. de soporte promedio	512	461	525	567
Tasa de error media	0.116	0.124	0.144	0.165

Tabla 4.3: Modas de los parámetros óptimos C y γ del modelo MVS con kernel Gaussiano, así como el número de células implementadas y su tamaño medio para cada periodo de la base de entrenamiento.

algoritmo de MVS posee el parámetro de costo C, por lo que para la calibración de los algoritmos en cada célula se usó una cuadrícula de todas las posibles parejas (γ, C) dentro de un rango de valores para ambos parámetros[22] y se escogió el modelo que minimizara la tasa de error estimada por 10-fold de validación cruzada.

La Tabla 4.3 muestra el número de células obtenidas para cada periodo y su tamaño medio, así como los valores de los parámetro óptimos con mayor frecuencia, y el número de vectores de soporte promedio en cada célula. Nótese que el número de células para 2010,

[22]La cuadrícula utilizada fue:
$(\gamma, C) \in \{0.01, 0.04, 0.25, 1, 4, 16, 100, 400\} \times \{0.01, 0.1, 1, 10, 100, 1000, 10000\}$.

2012, y 2014 se mantiene prácticamente constante, lo mismo que el tamaño medio de las células en 2012, 2014, y 2016. Intuitivamente, en 2016 se requieren un mayor número de células por tener una mayor dispersión en los datos, a diferencia de 2010 en donde simplemente se incrementó el tamaño medio de las mismas.

Con respecto a los parámetros óptimos, la mayoría de las células en 2010 y 2012 eligieron un valor óptimo para γ de 0.25, mientras que para 2014 y 2016 se redujo a 0.01. Los valores pequeños del parámetro γ están relacionados con un kernel Gaussiano de gran varianza, es decir, aunque dos observaciones se encuentren muy lejos una de la otra los valores pequeños de γ hacen que el kernel las identifique como similares. Asimismo, la moda del parámetro de costo C permanece sin cambios a través del tiempo con un valor igual a 10,000. Cuando C es grande, el margen se vuelve más ancho y se permite un mayor número de violaciones en él por parte de las observaciones. Además, el clasificador de MVS resultante es potencialmente más sesgado pero con menor varianza, tal y como lo menciona Tibshirani, James, Witten y Hastie en [17].

La tasa de error media de las células presentada en la Tabla 4.3 puede ser considerada como una estimación de la tasa de error del algoritmo. Obsérvese que el desempeño de las células de SVM es mejor significativamente a lo realizado por ADL en todos los periodos, y un poco mejor a lo obtenido con BA en 2010 y 2012. En tanto para 2014

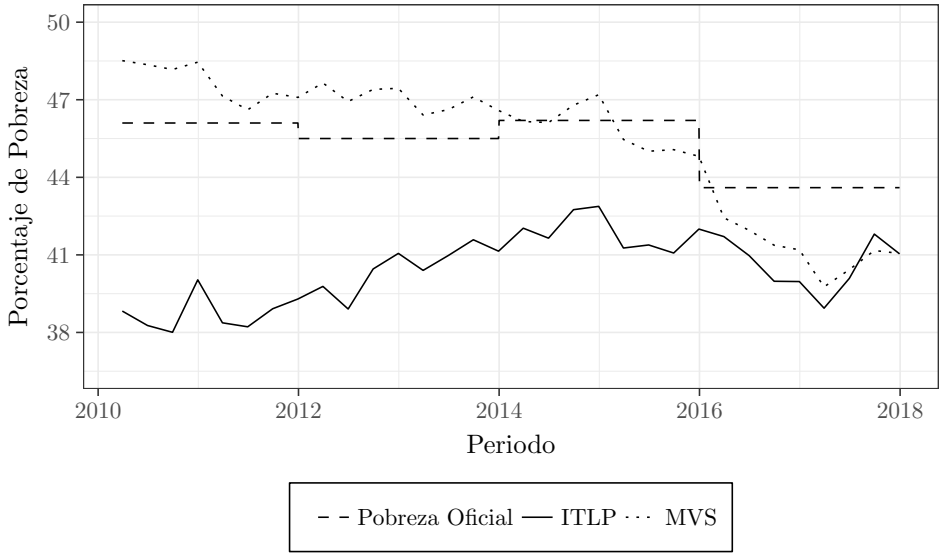

Figura 4.4: Pobreza laboral a nivel nacional estimada por el ITLP, Cifras Oficiales de Pobreza, y las células de MVS con un kernel Gaussiano, desde primer trimestre de 2010 hasta cuarto trimestre de 2017.

y 2016 es prácticamente igual a BA.

La Figura 4.4 muestra el indicador de la pobreza nacional predicha por el clasificador de las células de MVS. Es claro que la tendencia del enfoque alternativo de MVS no refleja la tendencia creciente del ITLP en la primera mitad de la ventana temporal, en cambio, sus cifras presentan una mayor precisión con respecto a las oficiales. Por otro lado, en la segunda mitad de la ventana temporal se puede apreciar que ambas tendencias coinciden, y que a partir de 2017 las proporciones son casi las mismas.

4.4. Desempeño de los algoritmos

El 27 de febrero de 2018 el Banco Mundial realizó una conferencia en sus instalaciones en Washington, D.C. titulada *Machine Learning and the Future of Poverty Prediction*.[23] En ella Olivier Dupriez, un estadista líder del Development Data Group, presentó una comparación empírica de diversos algoritmos de clasificación de Machine Learning aplicados a la predicción de la pobreza en Indonesia y Malaui. Una de las cosas más importantes de su presentación fue la de advertir que los resultados de las evaluaciones de los algoritmos de Machine Learning siempre se deben informar utilizando múltiples métricas de calidad, pues a pesar de que se tenga una buena predicción de la tasa de pobreza, no se garantiza que se tengan clasificados correctamente a los hogares pobres en el modelo. En sus propias palabras "*you might have the right numbers but not the right people...*".

Por dicha razón, se procedió a construir para cada periodo de la base de entrenamiento los modelos de ADL, BA y MVS (kernel Gaussiano) que maximizaran mediante validación cruzada (10-fold) las métricas de *Exactitud*, *Exhaustividad*, *Precisión*, *Valor* F_1, y *Kappa* (κ).[24] Es importante apuntar que ninguna métrica es mejor que

[23]Disponible en: https://www.worldbank.org/en/news/video/2018/02/27/machine-learning-future-of-poverty-prediction.

[24]Véase el Anexo D para una descripción más detallada sobre las métricas.

	Metricas de evaluación				
	Exactitud	Exhaustividad	Precisión	Valor F_1	κ
			2010		
ADL	0.763	0.815	0.758	0.786	0.522
BA	0.941	0.936	0.953	0.944	0.882
MVS	0.889	0.880	0.908	0.894	0.777
			2012		
ADL	0.744	0.789	0.743	0.765	0.484
BA	0.938	0.929	0.952	0.940	0.875
MVS	0.879	0.870	0.898	0.884	0.757
			2014		
ADL	0.721	0.818	0.711	0.761	0.431
BA	0.934	0.930	0.948	0.939	0.868
MVS	0.872	0.876	0.886	0.881	0.742
			2016		
ADL	0.711	0.874	0.706	0.781	0.371
BA	0.939	0.942	0.954	0.948	0.874
MVS	0.837	0.914	0.827	0.868	0.655

Tabla 4.4: Comparación de los modelos mediante diferentes métricas.

otra, simplemente evalúan conceptos diferentes y cada una aporta información distinta.

En este caso no se utilizaron cuadrículas para los parámetros de

ajuste de los algoritmos a diferencia de los procesos previos de calibración, y únicamente se trabajó con los valores predeterminados de las paqueterías del software estadístico *R*.

La Tabla 4.4 muestra los resultados obtenidos para las cinco métricas de evaluación empleadas y los distintos periodos de la base de entrenamiento. Realizando un análisis individual de los algoritmos, se puede identificar que en todos los años el mejor desempeño de ADL es bajo la métrica de Exhaustividad. Por el contrario, BA y MVS se desempeñan mucho mejor bajo la métrica de Precisión, a excepción de 2016 donde MVS realiza un mejor trabajo con la métrica de Exhaustividad.

Por otra parte, comparando a los algoritmos dentro de cada una de las métricas y periodos correspondientes, es fácil ver que los Bosques Aleatorios superan en cualquier escenario a los otros dos algoritmos de Machine Learning, confirmando su buen desempeño sobre fronteras de decisión altamente no lineales. Las Máquinas de Vectores de Soporte con kernel Gaussiano realizan una buena evaluación general y superan significativamente el desempeño del Análisis Discriminante Lineal. Interesantemente, el efecto de la dispersión de los datos en 2016 es prácticamente imperceptible bajo la evaluación conjunta de las distintas métricas utilizadas.

5
Conclusiones

Los métodos de Machine Learning, Análisis Discriminante Lineal, Bosques Aleatorios, y Máquinas de Vectores de Soporte, proporcionan un conjunto innovador de herramientas para la predicción de la pobreza en México en el corto plazo.

El desempeño individual de los algoritmos indica que ADL tiene la mayor tasa de error de los tres modelos utilizados, además de no reflejar la tendencia de la pobreza laboral del ITLP. Sin embargo, sus estimaciones de la pobreza multidimensional son más precisas que las del ITLP de 2010 a 2016. Por otro lado, el indicador de la pobreza laboral de BA reduce significativamente la tasa de error de ADL en todos los periodos y es consistente con la tendencia del ITLP, pero subestima aún más las cifras oficiales de la pobreza. El enfoque alternativo de las células de MVS con kernel Gaussiano muestra una tasa de error similar a la de BA y un incremento en la precisión de las estimaciones de la pobreza ofical con respecto a las cifras del ITLP, aunque no coincide con su tendencia en la primera mitad de la ventana temporal. Ciertamente, el indicador compuesto ITLP-BA es el

indicador de corto plazo que mejor estima las cifras de la pobreza multidimensional a nivel nacional sin perder la tendencia laboral de la pobreza del ITLP.

Por otra parte, la evaluación conjunta del desempeño de los algoritmos muestra que BA es el método de Machine Learning con las mejores medidas de Exactitud, Exhaustividad, Precisión, Valor F_1, y κ, para todos los periodos de la base de entrenamiento, seguido por MVS (kernel Gaussiano), y finalmente ADL. Reforzando así la robustez de BA como predictor, tal y como lo indica la literatura.

Es importante mencionar que el cambio en la metodología de la ENIGH 2016 afectó la dispersión de los datos de ese periodo, alterando el rendimiento de los algoritmos en 2016. Además, tal y como lo menciona Campos Vázquez en [3], existe una tendencia creciente en los últimos años a no reportar el ingreso en la ENOE, por lo que una alternativa para próximas investigaciones sería la implementación de métodos de imputación del ingreso en la base de prueba, con el fin de mejorar las estimaciones de los algoritmos de Machine Learning aquí planteados. Aunado a esta recomendación, la agregación de nuevos algoritmos, la homologación de variables de la vivienda, y la realización de un análisis de la pobreza a nivel entidad federativa, son caminos interesantes para futuras investigaciones.

Anexos

Anexo A
Estadísticas descriptivas

	Media y desviación estándar			
	2010	2012	2014	2016
Tamaño del hogar	3.807	3.713	3.721	3.665
	(1.925)	(1.891)	(1.855)	(1.841)
Horas trabajadas	71.52	71.69	72.60	76.63
	(53.04)	(53.14)	(52.78)	(54.27)
Ingreso laboral	7661.9	7992.6	8901.9	9371.0
	(11335.4)	(13393.9)	(21599.2)	(52960.6)
pob	0.370	0.400	0.392	0.349
	(0.483)	(0.490)	(0.488)	(0.477)
Observaciones	61,840	57,273	58,121	70,307

Tabla A.1: Estadísticas descriptivas a nivel hogar de la base de entrenamiento (ENIGH-MCS).

	Media y desviación estándar			
	2010	2012	2014	2016
Localidades rurales	0.254	0.267	0.255	0.376
	(0.435)	(0.443)	(0.436)	(0.484)
Sexo	0.512	0.511	0.511	0.511
	(0.500)	(0.500)	(0.500)	(0.500)
Edad	29.83	30.38	30.47	30.69
	(20.78)	(21.04)	(20.95)	(21.13)
PNEA	0.264	0.255	0.253	0.232
	(0.441)	(0.436)	(0.435)	(0.422)
PEA ocupada	0.402	0.421	0.425	0.455
	(0.490)	(0.494)	(0.494)	(0.498)
PEA desocupada	0.0273	0.0241	0.0217	0.0134
	(0.163)	(0.153)	(0.146)	(0.115)
PEA menor de 15	0.307	0.300	0.300	0.300
	(0.461)	(0.458)	(0.458)	(0.458)
Inasistencia a la escuela	0.695	0.699	0.695	0.698
	(0.460)	(0.459)	(0.461)	(0.459)
Primaria incompleta o menos	0.385	0.373	0.356	0.353
	(0.487)	(0.484)	(0.479)	(0.478)

Primaria completa o sec. incompleta	0.194	0.191	0.189	0.189
	(0.395)	(0.393)	(0.391)	(0.392)
Secundaria completa o mayor	0.421	0.436	0.455	0.458
	(0.494)	(0.496)	(0.498)	(0.498)
Carencia por rezago educativo	0.208	0.201	0.187	0.188
	(0.406)	(0.401)	(0.390)	(0.391)
Acceso al IMSS	0.277	0.256	0.276	0.284
	(0.447)	(0.436)	(0.447)	(0.451)
Acceso al ISSSTE	0.0754	0.0655	0.0607	0.0556
	(0.264)	(0.247)	(0.239)	(0.229)
Sin atención médica	0.271	0.194	0.170	0.140
	(0.444)	(0.396)	(0.376)	(0.347)
Pobreza	0.467	0.471	0.458	0.411
	(0.499)	(0.499)	(0.498)	(0.492)
Observaciones	235,423	212,674	216,209	257,647

Tabla A.2: Estadísticas descriptivas a nivel individual de la base de entrenamiento (ENIGH-MCS).

	Media y desviación estándar							
	2010				2011			
	I	II	III	IV	I	II	III	IV
Tamaño del hogar	3.764	3.762	3.756	3.732	3.718	3.704	3.701	3.691
	(1.869)	(1.881)	(1.876)	(1.850)	(1.852)	(1.846)	(1.839)	(1.845)
Horas trabajadas	62.53	63.20	62.88	62.73	62.29	61.98	62.18	63.53
	(49.10)	(49.41)	(50.01)	(49.20)	(49.08)	(49.08)	(49.30)	(49.74)
Ingreso laboral	7042.3	7156.3	7068.6	6782.0	7080.9	7162.6	7055.7	7027.0
	(8396.5)	(8236.4)	(8105.7)	(7867.2)	(7969.8)	(8152.6)	(8133.6)	(8294.8)
pob	0.348	0.343	0.343	0.364	0.351	0.349	0.353	0.357
	(0.476)	(0.475)	(0.475)	(0.481)	(0.477)	(0.477)	(0.478)	(0.479)
Observaciones	94,266	94,496	93,545	92,348	92,879	92,741	91,452	90,972

Tabla A.3: Estadísticas descriptivas a nivel hogar de la base de prueba (ENOE 2010 y 2011).

	Media y desviación estándar							
	2012				2013			
	I	II	III	IV	I	II	III	IV
Tamaño del hogar	3.695	3.681	3.673	3.655	3.663	3.653	3.658	3.678
	(1.840)	(1.834)	(1.830)	(1.818)	(1.813)	(1.809)	(1.808)	(1.812)
Horas trabajadas	62.94	62.64	63.54	63.13	61.56	62.30	62.42	64.51
	(49.17)	(49.11)	(49.84)	(49.22)	(48.71)	(48.65)	(49.13)	(49.38)
Ingreso laboral	7386.4	7512.8	7427.8	7283.6	7582.4	7565.3	7530.7	7556.6
	(8408.8)	(8462.0)	(9882.7)	(8919.5)	(9965.5)	(9005.0)	(8604.6)	(9036.0)
pob	0.361	0.356	0.368	0.378	0.369	0.370	0.373	0.370
	(0.480)	(0.479)	(0.482)	(0.485)	(0.483)	(0.483)	(0.484)	(0.483)
Observaciones	92,397	92,577	91,372	90,066	90,301	90,041	89,681	90,684

Tabla A.4: Estadísticas descriptivas a nivel hogar de la base de prueba (ENOE 2012 y 2013).

	Media y desviación estándar							
	2014				2015			
	I	II	III	IV	I	II	III	IV
Tamaño del hogar	3.671	3.649	3.638	3.634	3.619	3.613	3.601	3.597
	(1.796)	(1.789)	(1.783)	(1.783)	(1.780)	(1.776)	(1.764)	(1.770)
Horas trabajadas	61.84	61.20	62.02	63.39	62.25	62.09	62.51	62.92
	(48.50)	(47.57)	(48.34)	(48.30)	(48.07)	(48.47)	(48.72)	(48.54)
Ingreso laboral	7722.0	7619.4	7543.2	7483.5	7835.5	7930.4	7927.8	7897.5
	(8403.0)	(8234.8)	(8305.8)	(8028.8)	(8313.8)	(8489.3)	(8775.1)	(8569.7)
pob	0.377	0.376	0.385	0.381	0.368	0.364	0.366	0.377
	(0.485)	(0.484)	(0.487)	(0.486)	(0.482)	(0.481)	(0.482)	(0.485)
Observaciones	91,865	92,335	92,182	91,937	92,658	92,829	91,798	91,148

Tabla A.5: Estadísticas descriptivas a nivel hogar de la base de prueba (ENOE 2014 y 2015).

	Media y desviación estándar							
	2016				2017			
	I	II	III	IV	I	II	III	IV
Tamaño del hogar	3.577	3.566	3.540	3.540	3.526	3.515	3.509	3.506
	(1.763)	(1.756)	(1.751)	(1.760)	(1.755)	(1.762)	(1.758)	(1.753)
Horas trabajadas	60.90	63.05	62.59	63.01	62.26	60.81	61.58	62.23
	(47.98)	(48.01)	(48.77)	(48.44)	(48.21)	(47.79)	(48.31)	(48.34)
Ingreso laboral	8158.8	8234.1	8210.9	8151.2	8719.7	8642.4	8502.0	8471.1
	(8548.1)	(8787.1)	(8795.6)	(8677.6)	(9237.5)	(9299.3)	(9075.4)	(9043.3)
pob	0.375	0.367	0.365	0.363	0.359	0.366	0.381	0.375
	(0.484)	(0.482)	(0.481)	(0.481)	(0.480)	(0.482)	(0.486)	(0.484)
Observaciones	91,200	91,398	90,236	89,884	90,361	91,492	88,661	89,084

Tabla A.6: Estadísticas descriptivas a nivel hogar de la base de prueba (ENOE 2016 y 2017).

	Media y desviación estándar							
	2010				2011			
	I	II	III	IV	I	II	III	IV
Localidades rurales	0.191	0.192	0.191	0.194	0.194	0.193	0.193	0.198
	(0.393)	(0.394)	(0.393)	(0.395)	(0.395)	(0.394)	(0.394)	(0.398)
Sexo	0.517	0.516	0.516	0.516	0.516	0.515	0.516	0.517
	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)
PEA ocupada	0.398	0.406	0.403	0.397	0.397	0.403	0.403	0.411
	(0.490)	(0.491)	(0.491)	(0.489)	(0.489)	(0.490)	(0.490)	(0.492)
Carencia por rezago educativo	0.174	0.174	0.171	0.170	0.172	0.171	0.167	0.166
	(0.379)	(0.379)	(0.376)	(0.376)	(0.377)	(0.376)	(0.373)	(0.372)
Acceso al IMSS	0.318	0.319	0.322	0.329	0.320	0.322	0.320	0.321
	(0.466)	(0.466)	(0.467)	(0.470)	(0.466)	(0.467)	(0.467)	(0.467)
Sin atención médica	0.252	0.258	0.255	0.246	0.249	0.254	0.255	0.263
	(0.434)	(0.438)	(0.436)	(0.431)	(0.433)	(0.435)	(0.436)	(0.440)
Observaciones	354,833	355,449	351,313	344,659	345,337	343,474	338,431	335,765

Tabla A.7: Estadísticas descriptivas a nivel individual de la base de prueba (ENOE 2010 y 2011).

	Media y desviación estándar							
	2012				2013			
	I	II	III	IV	I	II	III	IV
Localidades rurales	0.195	0.194	0.194	0.198	0.192	0.188	0.181	0.176
	(0.396)	(0.395)	(0.396)	(0.398)	(0.394)	(0.390)	(0.385)	(0.381)
Sexo	0.516	0.516	0.516	0.517	0.516	0.516	0.516	0.516
	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)
PEA ocupada	0.406	0.414	0.415	0.409	0.405	0.411	0.408	0.413
	(0.491)	(0.493)	(0.493)	(0.492)	(0.491)	(0.492)	(0.491)	(0.492)
Carencia por rezago educativo	0.166	0.164	0.163	0.162	0.162	0.159	0.155	0.154
	(0.372)	(0.371)	(0.369)	(0.368)	(0.369)	(0.366)	(0.362)	(0.361)
Acceso al IMSS	0.319	0.322	0.325	0.330	0.327	0.334	0.337	0.343
	(0.466)	(0.467)	(0.468)	(0.470)	(0.469)	(0.472)	(0.473)	(0.475)
Sin atención médica	0.257	0.263	0.264	0.254	0.253	0.257	0.255	0.257
	(0.437)	(0.440)	(0.441)	(0.435)	(0.435)	(0.437)	(0.436)	(0.437)
Observaciones	341,394	340,803	335,605	329,204	330,745	328,944	328,059	333,492

Tabla A.8: Estadísticas descriptivas a nivel individual de la base de prueba (ENOE 2012 y 2013).

	Media y desviación estándar							
	2014				2015			
	I	II	III	IV	I	II	III	IV
Localidades rurales	0.169	0.170	0.172	0.171	0.171	0.171	0.172	0.172
	(0.375)	(0.376)	(0.377)	(0.377)	(0.377)	(0.377)	(0.377)	(0.378)
Sexo	0.516	0.516	0.516	0.518	0.517	0.517	0.516	0.516
	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)
PEA ocupada	0.405	0.406	0.406	0.409	0.409	0.412	0.411	0.416
	(0.491)	(0.491)	(0.491)	(0.492)	(0.492)	(0.492)	(0.492)	(0.493)
Carencia por rezago educativo	0.153	0.152	0.151	0.150	0.151	0.149	0.147	0.147
	(0.360)	(0.359)	(0.358)	(0.357)	(0.358)	(0.356)	(0.354)	(0.354)
Acceso al IMSS	0.349	0.350	0.350	0.349	0.350	0.353	0.351	0.350
	(0.477)	(0.477)	(0.477)	(0.477)	(0.477)	(0.478)	(0.477)	(0.477)
Sin atención médica	0.247	0.247	0.247	0.249	0.248	0.250	0.250	0.256
	(0.431)	(0.431)	(0.431)	(0.432)	(0.432)	(0.433)	(0.433)	(0.436)
Observaciones	337,258	336,915	335,376	334,130	335,374	335,362	330,572	327,820

Tabla A.9: Estadísticas descriptivas a nivel individual de la base de prueba (ENOE 2014 y 2015).

	Media y desviación estándar							
	2016				2017			
	I	II	III	IV	I	II	III	IV
Localidades rurales	0.172	0.173	0.173	0.176	0.175	0.171	0.173	0.173
	(0.378)	(0.378)	(0.378)	(0.381)	(0.380)	(0.376)	(0.378)	(0.378)
Sexo	0.516	0.517	0.518	0.517	0.518	0.517	0.517	0.516
	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)	(0.500)
PEA ocupada	0.412	0.416	0.419	0.419	0.416	0.417	0.415	0.418
	(0.492)	(0.493)	(0.493)	(0.493)	(0.493)	(0.493)	(0.493)	(0.493)
Carencia por rezago educativo	0.149	0.147	0.145	0.144	0.145	0.142	0.140	0.139
	(0.356)	(0.354)	(0.352)	(0.351)	(0.352)	(0.349)	(0.347)	(0.346)
Acceso al IMSS	0.347	0.351	0.350	0.354	0.350	0.354	0.353	0.353
	(0.476)	(0.477)	(0.477)	(0.478)	(0.477)	(0.478)	(0.478)	(0.478)
Sin atención médica	0.252	0.254	0.256	0.254	0.253	0.253	0.253	0.255
	(0.434)	(0.435)	(0.436)	(0.435)	(0.435)	(0.435)	(0.435)	(0.436)
Observaciones	326,265	325,948	319,394	318,179	318,578	321,559	311,112	312,352

Tabla A.10: Estadísticas descriptivas a nivel individual de la base de prueba (ENOE 2016 y 2017).

Fuente	Análisis de la varianza (ANOVA)				
	Suma de cuadrados	gl	Media cuadrática	F	p-valor
ENIGH-MCS 2010 y ENOE 2010 I - 2011 IV	3.06×10^{10}	8	3.83×10^{9}	53.78	0.0000
ENIGH-MCS 2012 y ENOE 2012 I - 2013 IV	2.09×10^{10}	8	2.62×10^{9}	29.38	0.0000
ENIGH-MCS 2014 y ENOE 2014 I - 2015 IV	9.28×10^{10}	8	1.16×10^{10}	116.70	0.0000
ENIGH-MCS 2016 y ENOE 2016 I - 2017 IV	9.46×10^{10}	8	1.18×10^{10}	36.78	0.0000

Tabla A.11: Prueba *one-way* ANOVA del ingreso medio entre la base de entrenamiento y la base de prueba, para los distintos periodos establecidos.

Anexo B
Pruebas de normalidad

	Clase *No Pobre*											
	2010			2012			2014			2016		
Prueba	Estadístico	χ^2	p-valor	Estadístico	χ^2	p-valor	Estadístico	χ^2	p-valor	Estadístico	χ^2	p-valor
Mardia *Skewness*	498.693	1.04×10^7	0.000	1268.645	2.38×10^7	0.000	16353.510	3.19×10^8	0.000	45320.070	1.15×10^9	0.000
Mardia *Kurtosis*	1433.339	6.27×10^8	0.000	3650.921	3.81×10^9	0.000	26399.470	2.12×10^{11}	0.000	52605.560	1.09×10^{12}	0.000
Henze-Zirkler	1029.588	5.19×10^5	0.000	1061.953	4.93×10^5	0.000	1442.844	5.49×10^5	0.000	1987.207	6.92×10^5	0.000
Doornik-Hansen	-	3.82×10^6	0.000	-	1.10×10^7	0.000	-	1.69×10^7	0.000	-	3.33×10^8	0.000

Tabla B.1: Resultados de las pruebas de Normalidad Multivariable de las variables normalizadas *edad*, *tamh*, *htrabh*, *ingıab* y sus interacciones utilizadas en ADL, para la clase *No Pobre* en todos los periodos de la base de entrenamiento.

	Clase *Pobre*											
	2010			2012			2014			2016		
Prueba	Estadístico	χ^2	p-valor	Estadístico	χ^2	p-valor	Estadístico	χ^2	p-valor	Estadístico	χ^2	p-valor
Mardia *Skewness*	309.333	5.67×10^6	0.000	235.743	3.93×10^6	0.000	126.134	2.08×10^6	0.000	145.280	2.56×10^6	0.000
Mardia *Kurtosis*	677.558	1.14×10^8	0.000	545.241	6.45×10^7	0.000	298.990	1.62×10^7	0.000	314.418	1.95×10^7	0.000
Henze-Zirkler	673.105	4.25×10^5	0.000	609.417	3.91×10^5	0.000	569.891	3.80×10^5	0.000	625.221	4.06×10^5	0.000
Doornik-Hansen	-	5.26×10^5	0.000	-	4.35×10^5	0.000	-	3.15×10^5	0.000	-	3.08×10^5	0.000

Tabla B.2: Resultados de las pruebas de Normalidad Multivariable de las variables normalizadas *edad*, *tamh*, *htrabh*, *ingl_ab* y sus interacciones utilizadas en ADL, para la clase *Pobre* en todos los periodos de la base de entrenamiento.

Anexo C

Índices de corto plazo de la pobreza

	Porcentaje de Pobreza							
	2010				2011			
	I	II	III	IV	I	II	III	IV
ITLP	38.83	38.27	38.01	40.03	38.37	38.22	38.92	39.29
ADL	49.34	48.57	48.66	49.08	47.55	47.31	47.44	47.04
BA	36.52	36.28	36.11	36.53	35.18	35.06	35.47	35.22
ITLP-BA	45.05	44.6	44.48	45.93	44.01	43.78	44.31	44.23
MVS	48.51	48.35	48.17	48.45	47.14	46.61	47.25	47.09
Oficial	46.11							

Tabla C.1: Cifras oficiales y predicciones de los algoritmos de Machine Learning de la tasa nacional de pobreza en los periodos 2010 y 2011.

	Porcentaje de Pobreza							
	2012				2013			
	I	II	III	IV	I	II	III	IV
ITLP	39.78	38.91	40.45	41.06	40.4	40.98	41.58	41.14
ADL	49.23	48.37	48.37	48.51	48.18	48.11	48.3	47.39
BA	37.06	36.26	36.81	36.76	35.63	36.18	36.61	36.09
ITLP-BA	46.5	45.46	46.44	46.73	45.65	46.28	46.96	46.42
MVS	47.65	46.93	47.4	47.44	46.41	46.62	47.11	46.58
Oficial	45.48							

Tabla C.2: Cifras oficiales y predicciones de los algoritmos de Machine Learning de la tasa nacional de pobreza en los periodos 2012 y 2013.

	Porcentaje de Pobreza							
	2014				2015			
	I	II	III	IV	I	II	III	IV
ITLP	42.03	41.65	42.75	42.88	41.27	41.38	41.07	42
ADL	46.1	46.59	46.36	46.03	45.16	44.88	43.96	43.82
BA	37.08	36.89	37.81	37.76	36.29	35.71	35.76	35.59
ITLP-BA	47.47	47.31	48.39	48.55	46.79	46.47	46.32	46.57
MVS	46.16	46.12	46.79	47.19	45.46	45.01	45.07	44.81
Oficial	46.17							

Tabla C.3: Cifras oficiales y predicciones de los algoritmos de Machine Learning de la tasa nacional de pobreza en los periodos 2014 y 2015.

	Porcentaje de Pobreza							
	2016				2017			
	I	II	III	IV	I	II	III	IV
ITLP	41.71	40.97	39.98	39.97	38.94	40.09	41.8	41.04
ADL	53.94	52.68	51.94	52.22	49.82	50.27	50.87	50.63
BA	31.02	30.62	30.06	29.96	28.5	28.97	29.59	29.55
ITLP-BA	44.66	43.88	43.08	43.02	41.12	42.19	43.55	43.17
MVS	42.43	41.95	41.37	41.2	39.76	40.43	41.17	41.06
Oficial	43.56							

Tabla C.4: Cifras oficiales y predicciones de los algoritmos de Machine Learning de la tasa nacional de pobreza en los periodos 2016 y 2017.

Anexo D
Métricas

En la literatura se sugiere que se utilice más de una métrica para evaluar el desempeño de los algoritmos de clasificación de Machine Learning. Dentro de las más comunes se encuentran aquellas basadas en los resultados de la *Matriz de confusión*, construida a partir de los valores predichos por el algoritmo y los valores reales de clasificación, tal y como se muestra en la siguiente tabla:

		Predicha	
		Pobre	No Pobre
Verdadera	Pobre	VP	FN
	No Pobre	FP	VN

Tabla D.1: Matriz de confusión.

Donde VP se refiere a los Verdaderos Positivos, FN a los Falsos Negativos, FP a los Falsos Positivos, y VN a los Verdaderos Negativos. Así pues, se definen las métricas de *Exactitud*, *Exhaustividad*, *Precisión*, *Valor* F_1, y *Kappa* (κ) de la siguiente manera:

$$Exactitud = \frac{VP + VN}{VP + FN + FP + VN}$$

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

$$Precisión = \frac{VP}{VP + FP}$$

$$Valor\ F_1 = 2\,\frac{Precisión \times Exhaustividad}{Precisión + Exhaustividad}$$

$$\kappa = \frac{Exactitud - E(Exactitud)}{1 - E(Exactitud)}$$

donde

$$\begin{aligned} E(Exactitud) =& P(\text{Verdadera} = \text{No Pobre}) \times P(\text{Predicha} = \text{No Pobre}) \\ &+ P(\text{Verdadera} = \text{Pobre}) \times P(\text{Predicha} = \text{Pobre}) \end{aligned}$$

Intuitivamente, la Exactitud refleja el porcentaje de predicciones correctas (de todas las clases posibles) realizadas por el modelo. Nótese que la tasa de error de un algoritmo es igual a $1 - Exactitud$. Por otro lado, la Exhaustividad muestra la proporción de aciertos dentro de la clase relevante (*Pobre*) verdadera, mientras que la Precisión indica la proporción de aciertos dentro de la clase relevante predicha por el

algoritmo. La media armónica entre la Precisión y la Exhaustividad da como resultado el valor F_1. Finalmente, Kappa es una medida de cuán cercanas están las observaciones clasificadas por el algoritmo con las etiquetas verdaderas, controlando por la Exactitud esperada de un clasificador aleatorio. Cabe señalar que todas las métricas son óptimas cuando son iguales a uno, y que a excepción de Kappa, se encuentran entre cero y uno.

Bibliografía

[1] Breiman, L. Random Forests. *Machine Learning 45*, 1 (2001), 5–32.

[2] Camara de Diputados de H. Congreso de la Union. Ley General de Desarrollo Social. *Diario Oficial de la Federación* (2004), 1–23.

[3] Campos-Vázquez, R. M. Efectos de los ingresos no reportados en el nivel y tendencia de la pobreza laboral en México. *Serie documentos de trabajo del Centro de Estudios Económicos, El Colegio de México, A. C.* (2013), 1–31.

[4] Chang, C.-c., and Lin, C.-j. LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST) 2* (2013), 1–39.

[5] Chih-Wei Hsu, Chih-Chung Chang, and Lin, C.-J. A Practical Guide to Support Vector Classification. *BJU international* (2016), 1–16.

[6] CONEVAL. Análisis de la evolución de la información de los ingresos laborales en la Encuesta Nacional de Ocupación y Empleo (ENOE). 1–5.

[7] CONEVAL. Anexo técnico para la construcción del Índice de la Tendencia Laboral de la Pobreza (ITLP). 1.

[8] CONEVAL. Tendencias económicas y sociales de corto plazo y el Índice de la tendencia laboral de la pobreza (ITLP). 1–49.

[9] CONEVAL. *Metodología para la medición multidimensional de la pobreza en México*, segunda ed ed. México, D.F, 2014.

[10] CONEVAL. CONEVAL Informa la Evolución de la Pobreza 2010-2016. 1–15.

[11] CONEVAL. Información referente al Índice de tendencia laboral de la pobreza al cuarto trimestre de 2017. 1–7.

[12] Cruz, J. A., and Wishart, D. S. Applications of machine learning in cancer prediction and prognosis. *Cancer Informatics 2* (2006), 59–77.

[13] Doornik, J. A., and Hansen, H. An omnibus test for univariate and multivariate normality. *Oxford Bulletin of Economics and Statistics 70*, SUPPL. 1 (2008), 927–939.

[14] Guo, G., Li, S. Z., and Chan, K. L. Support vector machines for face recognition. *Image and Vision Computing 19*, 9-10 (2001), 631–638.

[15] Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning*, vol. 2. 2009.

[16] Henze, N., and Zirkler, B. A class of invariant consistent tests for multivariate normality. *Communications in Statistics - Theory and Methods 19*, 10 (1990), 3595–3617.

[17] James, G., Witten, D., Tibshirani, R., and Hastie, T. *An Introduction to Statistical Learning with Applications in R*. Springer Texts in Statistics, New York, 2013.

[18] Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, S. M. Human Decisions and Machine Predictions. *The Quarterly Journal of Economics*, January (2018), 237–293.

[19] Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J., and Mullainathan, S. Human Decisions and Machine Predictions*. *The Quarterly Journal of Economics* (2017).

[20] Lantz, B. *Machine Learning with R - Second Edition*. 2015.

[21] Mardia, K. V. Measures of multivariate skewness and kurtosis with applications. *Biometrika 57*, 3 (1970), 519–530.

[22] McBride, L., and Nichols, A. Retooling Poverty Targeting Using Out-of-Sample Validation and Machine Learning. *The World Bank Economic Review* (2016), lhw056.

[23] Mullainathan, S., and Spiess, J. Machine Learning: An Applied Econometric Approach. *Journal of Economic Perspectives 31*, 2 (2017), 87–106.

[24] Sohnesen, T. P., and Stender, N. Is Random Forest a Superior Methodology for Predicting Poverty? An Empirical Assessment. *Poverty and Public Policy 9*, 1 (2016), 118–133.

[25] Steinwart, I., and Thomann, P. liquidSVM: A fast and versatile svm package. *ArXiv e-prints 1702.06899* (feb 2017).