Universidad de
SanAndrés

Big Data

Profesora: Noelia Romero
Asistente: Victoria Oubiña

Trabajo Práctico 4

Armas Braithwaite, Fernández, Menta, Vargas Ochuza

Fecha: 26/11/2023

Parte I: Análisis de la base

Inciso 3

Para comenzar, identificamos las variables con *missings* (NaN), y cuantificamos cuantas observaciones de esta característica contenían. Luego, establecimos el umbral para determinar qué columnas eliminar. El umbral aquí es el número mínimo de valores no-NaN que debe tener la columna para no ser eliminada. Aquí, decidimos eliminar las columnas con más de 3500 valores NaN.

Una vez eliminadas las variables que contenían más de 3500 *missings*, volvimos a identificar las variables con NaN, y cuantificarlos. Debido a que eran una baja cantidad de observaciones en las variables CH08, y P47T.

Es importante señalar que cuando nos enfrentamos a la presencia de *missing values*, implícitamente estamos enfrentándonos al *trade-off* sesgo varianza, donde podemos eliminar las variables que contienen NaN, a costa de poder perder una buena variable explicativa en el modelo, afectando el sesgo. Como también, si en lugar de eliminar la variable, eliminamos las observaciones, estamos perdiendo *statistical power*, afectando así a la varianza en el model.

Luego, para las variables: 'ITF_hogar', 'ITF_indv', 'IPCF_hogar', 'IPCF_indv', 'P21', y 'P47T', variables referidas a los ingresos de las observaciones, tanto a nivel individual como a nivel hogar, eliminamos los *outliers*. Estos los calculamos como aquellas observaciones que se encontraban en el 3 % derecho de la distribución, es decir, los más ricos.

A su vez, eliminamos aquellas columnas que contienen string y no aportan al análisis, y aquellas variables que tienen los mismos valores para todas las observaciones debido a que pueden generar multicolinealidad y no aportan variabilidad.

También, transformamos las siguientes variables categóricas enteras, que consideramos relevantes, en dummies: 'NIVEL_ED', 'CH07', 'CH08', 'CH09', 'CH12', 'CH15', 'CH16', 'ESTADO', 'CAT_OCUP', y 'CAT_INAC'.

Finalmente, eliminamos aquellas observaciones que reportaban ingresos negativos, y aquellas que informaban valores NaN carácterísticos de la EPH como los que reportan 99.

Como conclusión, luego de la limpieza calculamos las dimensiones de la base de datos. Pasamos de 263 variables y 7619 observaciones que conteníamos originalmente a 198

variables y 6360 observaciones.

Inciso 4

Para mejorar la predicción de la pobreza, inicialmente decidimos seguir las indicaciones del enunciado y crear un índice que refleje la proporción de niños presentes en el hogar. Creemos que esta variable es relevante para predecir la pobreza, ya que los niños generan gastos adicionales al ser personas adicionales en el hogar, lo que implica la necesidad de ingresos familiares más altos. Además, requieren tiempo y atención que podría destinarse a otras actividades productivas. Es importante considerar que los niños no contribuirán al hogar hasta alcanzar una edad determinada.

Como segundo conjunto de variables, optamos por incluir una serie de indicadores relacionados con la educación del jefe del hogar y su capital humano. El nivel de capital humano del jefe de familia puede ser un indicador sólido del nivel económico al que pueden acceder los miembros de la familia. Este factor no solo establece la base para acceder a trabajos mejor remunerados, sino que también sugiere que si el jefe de familia carece de oportunidades, es probable que su familia también las carezca. El conjunto de variables está específicamente relacionado con la educación incompleta en primaria, secundaria y universitaria del jefe del hogar. Cada uno de estos niveles representa un acceso diferenciado a oportunidades. Por ejemplo, si el jefe de familia tiene educación primaria incompleta, es más probable que se encuentre en situación de pobreza.

Inciso 5

En este apartado, exploramos la relación entre la proporción de niños y los ingresos del hogar. Se anticipa que los ingresos del hogar disminuyan en relación con un mayor cociente de niños respecto a adultos, debido a las razones que previamente hemos expuesto. Consideramos que este cociente solo ofrece información relevante para el nivel de ingresos cuando el número de niños en el hogar es significativamente alto, descartando valores menos significativos. Considerar ratios altos de niños respecto a adultos en la relación con los ingresos del hogar es relevante debido a dos factores: la demanda aumentada de cuidado reduce el tiempo para actividades remuneradas, y la limitación en la inversión en educación y desarrollo profesional afecta las oportunidades laborales y los ingresos a largo plazo. Enfocarse en estos ratios destaca cómo el cuidado de los niños impacta el tiempo disponible y la inversión en el desarrollo de los adultos,

influyendo en los ingresos y el progreso a largo plazo del hogar. Esta perspectiva revela una tendencia negativa entre los ingresos del hogar y la proporción de niños, permitiéndonos identificar patrones significativos en dicha relación para la predicción de la pobreza.

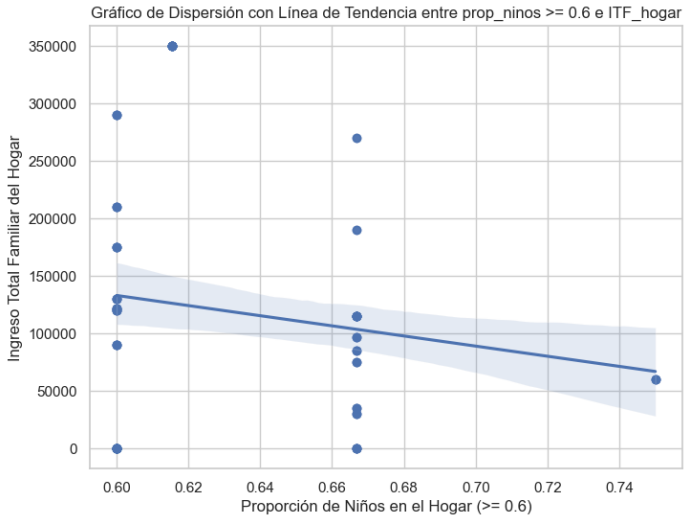

Figura 1: Relación entre prop_ninos > 0.6 e ITF_hogar

Inciso 6

El procedimiento fue realizado en el código. Una vez ejecutado el código, la cantidad de personas que dentro de la base respondieron que se encuentran por debajo del ingreso necesario es de 1.486, un 47 % de la base.

Inciso 7

En el presente inciso calculamos la tasa de hogares bajo la línea de pobreza utilizando una sola observación por hogar y sumando el ponderador PONDIH que permite expandir la muestra de la EPH al total de la población que representa.

Para nuestro modelo, la tasa de hogares bajo la línea de pobreza para el GBA es de 36.66 % cuantificando una cantidad de 1.439.914 pobres. La tasa de pobreza nos dio 6 p.p. mayor que la reportada por el INDEC. Creemos que esto puede ser debido a distintos criterios de limpieza de la base. Aun así, la diferencia no es significativa.

Parte II: Construcción de funciones

Evalua método

Esta función evalúa un modelo de clasificación. Especificamente, proporciona métricas clave y visualiza la curva ROC. También puede proveer los coeficientes estimados en el modelo y la proporción de coeficientes iguales a cero.[1] Para proveer lo anterior, esta función toma como insumo obligatorio al modelo de clasificación, los datos de entrenamiento, los datos de prueba. Además, se puede precisar si se desea visualizar o no la curva ROC y si se desea obtener los coeficientes del modelo y la proporción de coeficientes estimados iguales a cero.

Cross validation

Esta función realiza validación cruzada k-fold y evalúa el modelo de clasificación. Brinda un *dataframe* con k filas, una para cada iteración de la validación cruzada. Cada fila cuenta con el número de partición y el Error Cuadrático Medio. También se puede solicitar que se reporte la proporción de variables con coeficientes estimados iguales a cero.

Tiene como insumo el modelo de clasificación (inclusive, sus hiperparámetros), el número de particiones deseado y las bases de datos de la muestra de entrenamiento. Además, estandarizamos las variables como *default*. Esto es particularmente útil para evaluar modelos que requieren estandarización, como la regresión logística regularizada con Lasso. Esta función toma como insumo a la función *evalua metodo*.

Evalua config

Esta función evalúa la configuración del modelo de regresión logística y de K-vecinos cercanos. En el primer caso, obtiene el valor óptimo de λ bajo Ridge, Lasso o ambos métodos y ,en el segundo, el número óptimo de vecinos. El valor óptimo se define como el valor que minimiza el ECM promedio proveniente de realizar la validación cruzada con *cross validation*. Como insumo, debemos señalar la lista de hiperparámetros a evaluar, la muestra de entrenamiento y el número de particiones para la validación

[1] Esto presupone que el usuario utiliza la función para evaluar modelos que estiman coeficientes, como la regresión lógistica o la regresion lineal.

cruzada.

Evalua multiples métodos

Esta función evalúa varios métodos de clasificación. Se optimiza la elección de hiperparámetros para todos los métodos menos ADL. Puntualmente, se consideran los siguientes métodos:

- Regresión logística
- Analisis de discriminante lineal
- KNN
- Árbol de decisión
- Bagging
- Random Forests
- Ada Boosting

Toma como insumo las bases de datos de las muestras de entrenamiento y de prueba al igual que una lista de diccionarios que especifiquen los hiperparámetros a evaluar para cada método. Puntualmente, la salida de la función es un *dataframe* que contiene ,para cada método listado, las métricas de predicción evaluadas, los hiperparámetros considerados y los seleccionados como óptimos. Además, para el caso de la regresión logística, se señala el método de regularización óptimo.

Utilizamos la función *evalua config* para encontrar la combinación de método de regularización - costo de complejitud óptima para la Regresión Logística y para encontrar el número óptimo de vecinos para K-vecinos cercanos (KKN). Con respecto a los métodos basados en árboles, utilizamos la función *Grid Search CV* para hallar la combinación óptima de hiperparámetros para cada método.

Vale precisar que primero encontramos la configuración óptima para cada método y luego comparamos las métricas de cada método bajo sus hiperparámetros óptimos con la función *evalúa método*. El proceso de determinación de la configuración óptima utiliza la muestra de entrenamiento y la particiona, mientras que la comparación de métricas entre modelos genera las métricas con la muestra de prueba. De esta manera, evaluamos la capacidad predictiva *out of sample* de cada modelo.

Parte III: Clasificación y regularización

Inciso 1

En esta sección predecimos el porcentaje de individuos y hogares bajo la línea de pobreza en la muestra de encuestados que no indican su nivel de ingreso.

Para ello, en primer lugar, eliminamos de la base respondieron y no_respondieron todas las variables asociadas al ingreso. A su vez, también eliminamos las variables adulto_equiv, ad_equiv_hogar e ingreso_necesario, tal como solicita el enunciado. Luego, establecimos a la variable pobre como su variable dependiente, y al resto de las variables como independientes.

En segundo lugar, utilizaremos la muestra que se encuentra en la base respondieron para elegir el modelo óptimo en términos de capacidad de predicción *out of sample*.

Inciso 2

Luego evaluaremos todos los modelos incluidos en la función *evalúa multiples métodos*. Para todas las metodologías excepto por Análisis Discriminante Lineal, elegimos una combinación de hiperparámetros que minimizan el Error Cuadrático Medio a partir de un conjunto de hiperparámetros posibles. En el caso de la regresión lógistica, se elige la combinación óptima de método de regularización y costo de complejitud (λ) y ,en el caso de K-vecinos cercanos, se selecciona el número de vecinos óptimo. Asimismo, para el árbol de decisión se elige la profundidad óptima; para *bagging*, el número de árboles (*n_estimators*) y el tamaño de muestra en cada repetición (*max_samples*); para *random forest*, lo mismo que para *bagging*; finalmente, para *adaboosting*, el número de árboles y el ratio de aprendizaje. La Tabla 1 resume los resultados de esta evaluación de modelos.

Evaluación de modelos

Modelo	**AUC**	**Confusion Matrix**	**Accuracy Score**	**ECM**	**Hiperparámetro Óptimo**
Regresión Logística	0.77	[[404, 117], [100, 321]]	0.77	0.23	{C=1,penalty='l2'}
Análisis de Discriminante Lineal	0.76	[[413, 108], [112, 309]]	0.77	0.23	–
KNN	0.75	[[398, 123], [111, 310]]	0.75	0.25	{'n_neighbors':5}
Árbol de decisión	0.81	[[431, 90], [84, 337]]	0.82	0.18	{'max_depth': 20}
Bagging	0.88	[[469, 52], [55, 366]]	0.89	0.11	{'max_samples': 1.0, 'n_estimators': 400}
Random Forest	0.86	[[469, 52], [74, 347]]	0.87	0.13	{'max_depth': None, 'n_estimators': 400}
Ada Boost	0.81	[[423, 98], [81, 340]]	0.81	0.19	{'learning_rate': 1.0, 'n_estimators': 500}

Notas: Usamos la notación de parámetros del paquete *scikit-learn*. La lista completa de hiperparámetros evaluados se encuentra en el código.

Inciso 3

Luego de evaluar las métricas proporcionadas para cada modelo, concluimos que el modelo de Bagging muestra el mejor desempeño en general, basádonos en las métricas utilizadas:

- Posee el área bajo la curva (AUC) más alto, alcanzando 0.887426, lo que sugiere una excelente capacidad para distinguir entre las clases.
- Su precisión general, medida por el Accuracy Score, es alta (0.89), lo que indica un buen rendimiento en la clasificación correcta de las muestras. Además, el ECM es el mas bajo, reportando un valor de 0.11, lo que refleja precisión en las predicciones.
- Al analizar la Confusion Matrix, observamos un bajo número de falsos positivos y falsos negativos, lo que confirma la capacidad del modelo para predecir con precisión ambas clases.

En contraste, otros modelos como Análisis de Discriminante Lineal, KNN, y Ada Boost tienen métricas ligeramente más bajas en comparación con el modelo de Bagging en términos de AUC y precisión general.

Como ya dijimos, el modelo de Bagging con hiperparámetros de $max\_samples = 1$ y $n\_estimators = 400$, parece ser el más sólido y equilibrado en su rendimiento predictivo. Esto significa que se generan 400 árboles de decisión y que se utiliza la muestra completa en cada repetición.

Inciso 4

Las predicciones ahora tienen una precisión notablemente mayor. Hubo un aumento de más del 10 PP en la métrica de precisión, y se observó una mejora en casi todas las demás métricas con la utilización del modelo Bagging. Esto se encuentra relacionado con que en la Parte 1 - Inciso 4, introdujimos dos variables relevantes para el análisis, la proporción de niños en el hogar, y el nivel de capital humano del jefe de familia. Además, el hecho de haber considerado métodos en base a árboles de decisión parece haber contribuido, dado que en general poseen mejores resultados en las métricas de predicción.

Inciso 5

A partir del método Bagging, predecimos que el 55.38 % de las personas, dentro de la

base norespondieron, son pobres. Cuantificando una cifra de 31.557 personas.