Universidad de
SanAndrés

Big Data

Trabajo Práctico Grupal

Samuel Arispe, Gonzalo Ochoa y Facundo Valle Quintana

27 de noviembre de 2023

Trabajo Práctico 4

Big Data

Parte I: Análisis de la base de hogares y cálculo de pobreza

Ahora que ya se han familiarizado con la Encuesta Permanente de Hogares (EPH), asegúrense de estar calculando niveles de pobreza a nivel individual y a nivel de hogar similares a los que calcula el INDEC para la misma región. Además, asegúrense de estar realizando una limpieza de la base con mayor dedicación y entendimiento de sus variables.

1. Descarguen la base de microdatos de la EPH correspondiente al primer trimestre de 2023 (la base de hogares se llama *usu_hogar_T123.xls*). Importen los datos de la encuesta de hogar y, al igual que en los trabajos anteriores, conserven sólo las observaciones que corresponden a los aglomerados de Ciudad Autónoma de Buenos Aires o del Gran Buenos Aires.

Mediante un operador booleano (*.isin*), filtramos la base de datos por aglomerado, quedandonos únicamente con los datos pertenecientes a los aglomerados 32 y 33, que corresponden a la CABA y partidos del GBA respectivamente.

2. Unan la tabla de la encuesta individual con la de la encuesta de hogar. Asegúrense de estar usando las variables *CODUSU* y *NRO_HOGAR*.

Mediante *pd.merge* procedimos a unir las bases de datos.

3. Limpien la base de datos tomando criterios que hagan sentido, tanto para el tratamiento de valores faltantes, de *outliers*, como así también decidan qué variables categóricas y strings usarán y transfórmenlas de forma que haga sentido para los ejercicios siguientes. Justifiquen sus decisiones.

Borramos todas las variables que tienen más del 50 % de *missings*. Todas las variables relacionadas a los desocupados tienen más del 70 % de NaN mientras que todas las variables relacionadas a los ocupados tienen mas del 50 % de NaN. También eliminamos las variables relacionadas a la categoría 'otros' y $CH14$ que se refiere al último año que aprobó, que es redundante dado que ya tenemos nivel educativo ($NIVEL\_ED$).

Vemos que $CH06$ (Edad) tiene valores negativos, por lo tanto procedemos a eliminar esas observaciones. Así como las observaciones que tienen 99 en variables numéricas discretas que claramente son *missing* a excepción de la variable edad que puede tener el valor 99.

Convertimos a variables categóricas todas estas variables que son discretas y cuyo orden no quiere decir algo. Además al volverlas categóricas podemos identificar los *missing values* y podemos capturarlos de acuerdo a cada variable: $IV1$, $IV2$, $IV3$, $IV4$, $IV5$, $IV6$, $IV7$, $IV8$, $IV9$, $IV10$, $IV11$, $IV12\_1$, $IV12\_2$, $IV12\_3$, $II3$, $II4\_1$, $II4\_2$, $II4\_3$, $II5$, $II6$, $II7$, $II8$, $II9$, $V1$, $V2$, $V21$, $V22$, $V3$, $V4$, $V5$, $V6$, $V7$, $V8$, $V9$, $V10$, $V11$, $V12$, $V13$, $V14$, $V15$, $V16$, $V17$, $V18$, $V19\_A$, $V19\_B$, $CH03$, $CH04$, $CH07$, $CH08$, $CH09$, $CH10$, $CH11$, $CH13$, $CH15$, $CH16$, $ESTADO$, CAT_OCUP, CAT_INAC, $PP02C1$, $PP02C2$, $PP02C3$, $PP02C4$, $PP02C5$, $PP02C6$, $PP02C7$, $PP02C8$, $PP02E$, $PP02H$, $PP02I$

4. Construyan variables (mínimo 2) que no estén en la base pero que sean relevantes para predecir individuos bajo la línea de pobreza (por ejemplo, la proporción de niños en el hogar, si el cónyuge trabaja).

Procedemos a construir dos variables:

1. *Menores_hogar*: la proporción de niños en el hogar, que resulta de la división entre la cantidad de menores encuestados IX_Men10 y la cantidad de miembros del hogar IX_TOT.
2. *Pareja_trabaja*: un indicador binario de si el cónyuge se encuentra en actividad: tomando el valor 1 cuando las condiciones $CH03$ es igual a 2 (se trata de un cónyuge) y $ESTADO$ vale 1 (se encuentra en actividad) y 0 (indicando Falso) en caso contrario.

.

5. Presenten un gráfico (que no sea de barras) para describir la interacción o correlación entre dos o más variables.

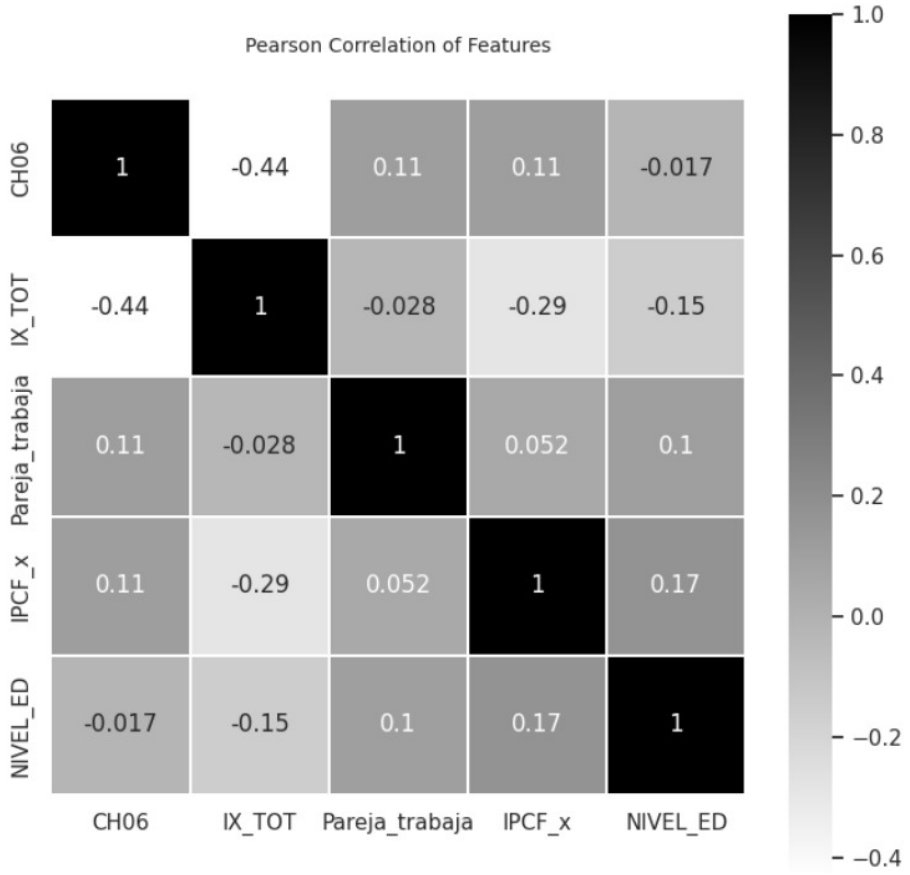

Figura 1: Matriz de correlaciones entre variables seleccionadas

Realizamos una correlación entre las variables '$CH06$', 'IX_TOT', '$Pareja\_trabaja$', '$IPCF\_x$', '$NIVEL\_ED$'.

Podemos ver que una correlación negativa entre la cantidad total de miembros del hogar (IX_TOT) y y la edad de los encuestados, lo que tiene sentido económico.

Otra correlación interesante es la de Ingreso Capital per Capita Familiar ($IPCF\_x$) y la cantidad de miembros del hogar (IX_TOT), lo que una vez más tiene sentido (algebraícamente).

6. Construyan la columna *_adulto_equiv* y la columna *ad_equiv_hogar* y luego dividan la base en dos dataframes donde: uno conserve las personas que no reportaron ITF (dataframe llamado respondieron) y otro conserve a las personas que no reportaron ITF (llamado norespondieron). Además, agreguen a la base respondieron una columna llamada *ingreso_necesario* que sea el producto de la canasta básica por *ad_equiv_hogar*. Agreguen a la base respondieron una columna llamada *pobre*,

que tome valor 1 si el ITF es menor al *ingreso_necesario* que necesita esa familia y 0 en caso contrario.

Procedimos a crear las variables requeridas. Volvemos a emplear nuestro código del TP2.

7. Para calcular la tasa de hogares bajo la línea de pobreza utilicen una sola observación por hogar y sumen el ponderador PONDIH que permite expandir la muestra de la EPH al total de la población que representa. ¿Cuál es la tasa de hogares bajo la línea de pobreza para el GBA? ¿Lograron que se asemeje al porcentaje que reporta el INDEC en sus informes?

Los resultados muestran que, a nivel de hogares, el 71.53 % de los hogares no se consideran "pobres", mientras que el 28.47 % de los hogares se consideran "pobres".

El INDEC en su informe, señala que el porcentaje de hogares bajo la linea de pobreza para el GBA es del 30.3 %, indicando que nuestra estimación de pobreza para el GBA resulta bastante acertada.

Parte II: Construcción de funciones

El objetivo de esta parte del trabajo es revisar y mejorar el código que escribieron en la parte II del TP3. Deben buscar que sea flexible y esté modularizado (en funciones bien documentadas con docstrings). De esta forma, evitarán repetir código y podrán utilizarlo en distintos escenarios (como por ejemplo la Parte III de este TP y sus proyectos personales a futuro).

1. Escriban una función, llamada *evalua_metodo*, que reciba como argumentos un modelo y los datos de entrenamiento y prueba (*X train*,*y train*, *X test*, *y test*). La función debe ajustar el modelo con los datos de entrenamiento y calcular las métricas que considere necesarias para esta problemática (de mínima, deben reportar verdaderos positivos, verdaderos negativos, falsos positivos, falsos negativos, AUC, accuracy y precision de cada método). El output de la función debe ser una colección con las métricas evaluadas.

Procedimos con desarrollar el código considerando, a diferencia del TP3, las clasificaciones de CART además de las del TP3 (agregándolas).

2. Escriban una función, llamada *cross_validation*, que realice validación cruzada con k iteraciones (k-fold CV), llamando a la función del inciso anterior en cada una, pero para las k distintas particiones. La función debe recibir como argumentos el modelo, el valor de k y un dataset (es decir, sólo X e y [1] Pueden ayudarse con la función Kfold para generar las particiones necesarias.).

Procedimos a desarrollar el código considerando como input X_train y y_train y no la base total.

3. Escriban una función, llamada *evalua_config* que reciba una lista de configuraciones de hiperparámetros (los distintos valores a probar como hiperparámetros podrían codificarse en diccionarios de Python) y utilizando la función *cross_validation* obtenga el error [2] promedio para cada configuración. Finalmente, la función debe devolver la configuración que genere menor error. Asegúrense de que esta función sirva para cualquier hiperparámetro que quieran elegir por crossvalidation para cualquier modelo.

Procedimos a desarrollar el código pidiéndole que, según el modelo de clasificación, evalúe los siguientes parámetros:

[1] Cuando usen esta función en la parte III del TP asegurense de pasar *X_train* e *y_train* para no utilizar las observaciones de test en esta instancia de validación.

[2] Utilicen la medición del error que prefieran. Una opción sería el Error Cuadrático Medio

- *vecinos*: número de vecinos más cercanos en KNeighborsClassifier.
- *depth*: profundidad del árbol, cantidad de nodos en DecisionTreeClassifier.
- *b_estimators*: número de muestras en BaggingClassifier.
- $features$: cantidad de variables en RandomForestClassifier.

4. Escriban una función llamada *evalua_multiples_metodo* que les permita implementar los siguiente métodos que se enumeran a continuación. Esta función debe utilizar su función *evalua_config* para optimizar los parámetros que ustedes decidan (de mínima deben optimizar el K -cantidad de vecinos- para el modelo KNN). Finalmente, el output de la función debe ser una tabla donde las columnas sean las métricas que hayan evaluado (las que hayan incluido en la función *evalua_metodo* y las filas sean los modelos (con su configuración de hiperparámetros asociada) que hayan corrido. Asegúrense de que la tabla incluya una columna con nombre del modelo y el valor de los hiperparámetros/configuración [3]:

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

Parte III: Clasificación y regularización

1. Eliminen de ambas bases (*respondieron*, *norespondieron*) todas las variables relacionadas a ingresos (en el archivo Diseño de bases y estructura ver las categorías: ingresos de la ocupación principal de los asalariados, ingresos de la ocupación principal, ingresos de otras ocupaciones, ingreso total individual, ingresos no laborales, ingreso total familiar, ingreso per cápita familiar). Eliminen también las columnas *adulto_equiv*, *ad_equiv_hogar* e *ingreso_necesario*. Establezcan a la variable pobre como su variable dependiente (vector y). El resto de las variables serán las variables independientes (matriz X).

Procedimos a realizar lo que pide el enunciado.

2. Corran la función *evalua_multiples_metodos* con la base respondieron. Asegúrense de estar utilizando su función de *evalua_config* para optimizar algunos hiperparámetros (de mínima, el K en el modelo KNN).

Procedimos a realizar lo que pide el enunciado.

[3] Para los modelos que elijan optimizar sus hiperparámetros (ej. KNN), observen que deberán correr la función *evalua_metodo* dos veces. Una para optimizar los hiperparámetros (con el set de datos para train, que será dividido nuevamente en train y validación) y otra, para obtener las métricas con el hiperparámetro óptimo obtenido (con un set de datos para train y otro para test).

3. ¿Cuál de todos los métodos evaluados predice mejor? ¿Con qué hiperparámetros? Justifiquen mencionando las métricas que conocen.

Tabla 1: Resultados de los diferentes modelos de regresión y clasificación

Modelo	Hiperparámetro	Matriz de Confusión	AUC	Accuracy	Precisión	ECM
Regresión Logística	-	$\begin{bmatrix}702 & 92\\131 & 327\end{bmatrix}$	0.799	0.822	0.780	0.178115
K Vecinos más Cercanos	3	$\begin{bmatrix}720 & 74\\197 & 261\end{bmatrix}$	0.738	0.784	0.779	0.216454
Análisis Discriminante	1	$\begin{bmatrix}714 & 80\\142 & 316\end{bmatrix}$	0.795	0.823	0.798	0.177316
Árbol de Decisión	21	$\begin{bmatrix}716 & 78\\89 & 369\end{bmatrix}$	0.853	0.866	0.825	0.134185
Bagging	62	$\begin{bmatrix}714 & 80\\135 & 323\end{bmatrix}$	0.802	0.828	0.801	0.171725
Random Forest	39	$\begin{bmatrix}725 & 69\\155 & 303\end{bmatrix}$	0.787	0.821	0.815	0.178914
Boosting	-	$\begin{bmatrix}693 & 101\\143 & 315\end{bmatrix}$	0.780	0.805	0.757	0.194888

En nuestro caso, vemos que el modelo con mejor capacidad predictiva es el de Árbol de decisión con una profundidad de valor **21** con **AUC 85.3 %** y y un **accuracy de 86.6 %** y una **precisión de 82.5 %** y un **ECM de 0.13**.

Luego le sigue en Bagging con **62** muestras como el número óptimo de muestras, un ECM de 0.172 y un **accuracy de 82.8 %**.

4. ¿Lograron mejorar sus predicción respecto al TP3 [4]?

Sí, claramente hubo una mejora. En comparación con el TP3, podemos ver que nuestros métodos se volvieron en general más precisos:

Tabla 2: TP3

Modelo	Hiperparámetro	Matriz de confusión	Accuracy	AUC	ECM
Regresión logística Lasso	10.01	$\begin{bmatrix}703 & 91\\137 & 321\end{bmatrix}$	0.818	0.895	0.182109
K vecinos más cercanos	3.00	$\begin{bmatrix}714 & 80\\203 & 255\end{bmatrix}$	0.774	0.809	0.226038
Análisis discriminante	1.00	$\begin{bmatrix}719 & 75\\146 & 312\end{bmatrix}$	0.823	0.895	0.176518
Regresión Logística Ridge	100	$\begin{bmatrix}684 & 110\\107 & 351\end{bmatrix}$	0.827	0.896	0.173323

[4] Si crearon variables relevantes en el ejercicio 4 parte I, se esperaría que sus modelos mejoren su predicción.

Tabla 3: TP4

Modelo	**Hiperparámetro**	**Matriz de Confusión**	**AUC**	**Accuracy**	**Precisión**	**ECM**
Regresión Logística	-	$\begin{bmatrix} 702 & 92 \\ 131 & 327 \end{bmatrix}$	0.799	0.822	0.780	0.178115
K Vecinos más Cercanos	3	$\begin{bmatrix} 720 & 74 \\ 197 & 261 \end{bmatrix}$	0.738	0.784	0.779	0.216454
Análisis Discriminante	1	$\begin{bmatrix} 714 & 80 \\ 142 & 316 \end{bmatrix}$	0.795	0.823	0.798	0.177316
Árbol de Decisión	21	$\begin{bmatrix} 716 & 78 \\ 89 & 369 \end{bmatrix}$	0.853	0.866	0.825	0.134185
Bagging	62	$\begin{bmatrix} 714 & 80 \\ 135 & 323 \end{bmatrix}$	0.802	0.828	0.801	0.171725
Random Forest	39	$\begin{bmatrix} 725 & 69 \\ 155 & 303 \end{bmatrix}$	0.787	0.821	0.815	0.178914
Boosting	-	$\begin{bmatrix} 693 & 101 \\ 143 & 315 \end{bmatrix}$	0.780	0.805	0.757	0.194888

- Para la regresión logística, vemos que un leve empeoramiento en las métricas de versión estimada Ridge en el TP3, pero mejoras respecto a su versión Lasso. Esto solo por incorporar las nuevas variables creadas.
- Para el método de KNN, vemos leves mejoras en ECM y accuracy con el mismo hiperparámetro. Esto solo por incorporar las nuevas variables creadas.
- Para el el método LDA, vemos la misma precisión pero empeoramiento de la AUC y el ECM, pero la misma accuracy. Esto solo por incorporar las nuevas variables creadas.

5. Con el método que seleccionaron, predigan qué personas son pobres dentro de la base *norespondieron*. ¿Qué proporción de los hogares son pobres en esa submuestra?

La cantidad de pobres predicha en la muestra que no respondio es de: 1348 de 3387 . La tasa de pobreza predicha en la muestra de los que no respondieron es de: 39.8 %