Big Data y Machine Learning: Trabajo Práctico Nº2

Catalina Banfi Matías Lima Santiago López

1. Parte 1: Analizando la base

1.1. La medición de la pobreza en Argentina

En Argentina, el INDEC mide la incidencia de la pobreza utilizando un método de medición indirecta, por el cual se calcula una línea (esencialmente un umbral de ingresos) por debajo de la cual un hogar es considerado pobre. Esta línea se construye a partir del costo de adquirir la Canasta Básica Total, que se compone de la Canasta Básica de Alimentos (la cantidad mínima de necesidades energéticas y proteicas que debe consumir una persona) más la inclusión de bienes y servicios no alimentarios (como vestimenta, transporte, salud, etc.). Este último componente se construye en base a la evidencia empírica que refleja los hábitos de consumo de bienes y servicios no alimentarios de la población de referencia, mientras que la Canasta Básica de Alimentos es una canasta normativa. Como las necesidades nutricionales varían según la persona, el INDEC construye un índice llamado "adulto equivalente" que establece las relaciones en las necesidades energéticas según sexo y hogar a partir de un estándar que es el de un hombre adulto de actividad moderada (INDEC, 2016).

1.2.

Punto a

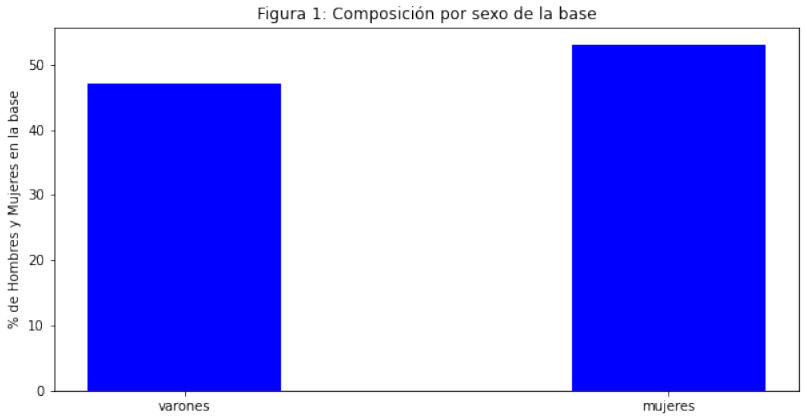
Naturalmente, lo primero que hacemos es importar la base a Python y luego nos quedamos solamente con los aglomerados urbanos 32 y 33 que corresponden a la Ciudad Autónoma de Buenos Aires y los Partidos del Grab Buenos Aires, respectivamente.

Punto b

Luego, borramos de la base todas las variables de ingreso que tengan valores negativos, así como todas las edades negativas, que representan respuestas que carecen de sentido práctico.

Punto c

Ya con la base limpia y filtrada en los aglomerados que queremos estudiar calculamos la proporción de hombres y mujeres en nuestra base, contando la cantidad de respuestas que se identificaron como uno o la otra. Encontramos que aproximadamente el 47 % de los encuestados se identificó como hombres, mientras que el restante 53 % se identificó como mujer. A continuación presentamos un gráfico de barras mostrando esta composición.



Punto d

Utilizando la librería *seaborn* construimos una matriz de correlaciones entre las variables sexo (”CH04”), estado civil (”CH07”), el tipo de cobertura médica que tiene el entrevistado (pública, privada, PAMI, etc.), o si tiene alguna (”CH08”), el nivel educativo que alcanzó (”NIVEL_ED), si tiene o no trabajo, está inactivo o tiene menos de 10 años (”ESTADO”), la razón de su inactividad, es decir, si es estudiante, jubilado, rentista, discapacitado, ama de casa, menor de 6 años o alguna otra razón (”CAT_INAC”) y, por último, el ingreso per cápita familiar (”IPCF”).

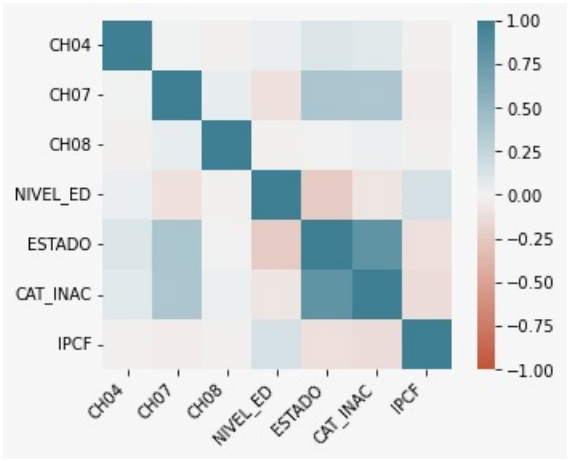


Figura 2: Matriz de Correlaciones

La escala del gráfico nos dice que cuanto más rojo un cuadrante (en términos de intensidad), mayor es la correlación negativa entre dos variables, y cuanto más azul, mayor es la correlación positiva. Por otro lado, mientras más claro un cuadrante, más cercana a 0 la correlación. Podemos ver, por ejemplo, que hay una correlación negativa significativa entre el nivel educativo y el estado ocupacional del entrevistado (INDEC asigna números más altos a desocupación e inactividad), lo que implica que a mayor nivel educativo[1] menores son las chances de estar desocupado o inactivo, aunque también aquí entran los menores de 10 años. Por otro ldo, como es de esperar, hay correlaciones negativas entre la inactividad y el ingreso per cápita, así como entre el estado ocupacional y el ingreso per cápita. Entre las correlaciones positivas podemos encontrar que el estado civil está correlacionado positivamente con la categoría de inactividad y el estado ocupacional. Esto tiene sentido

[1]Hay que tener presente que INDEC identifica con un 7 a quienes no recibieron instrucción y con un 9 a quienes o no responden o no saben, de todas formas dado que es natural esperar una correlación negativa entre estar empleado y el nivel educativo, estas observaciones no están sesgando los resultados

ya que el estado civil toma valores de 1 a 5 que significan unido, casado, separado, viudo y soltero. Mientras que la categoría de inactividad toma valores de 1 a 7 que representan si el encuestado es jubilado, rentista, estudiante, ama de casa, menor de 6 años, discapacitado u otros.

Punto e

En este punto queremos saber cuántos desocupados e inactivos hay en la muestra. Sabiendo que a los desocupados se los identifica con un 2 y a los inactivos con un 3 contamos la cantidad de filas que cumplen con cada una de estas condiciones en en la columna "ESTADO". Encontramos que en nuestra base hay 232 desocupados y 2695 inactivos. Luego, queremos saber la media del ingreso per cápita familiar de los ocupados (identificados con un 1), los desocupados e inactivos. Para esto partimos la base en tres partes y usamos la función *.mean()* que nos devuelve la media del IPCF de cada grupo. Obtenemos como es de esperar que los ocupados tienen la media de ingresos más alta con $31092.3, los inactivos la segunda más alta con $22358.18 y los desocupados la más baja con $14757.69.

Punto f

Para este punto importamos ahora la base de tabla de equivalencias que clasifica según edad y sexo a las personas en su relación al estándar del adulto equivalente. Para poder trabajarla en Python, al importarla le quitamos todas las filas que no tengan valores numéricos. Luego, para poder compatibilizarla con la base con la que venimos trabajando renombramos la fila que corresponden a las personas de 1 año (escrita originalmente como "1año" (sic)) a 1 "1 años". Luego, en nuestra base original creamos una columna nueva llamada "Edad" en la que, utilizando una función lambda y ciertas condiciones en base a la columna "CHO6" que declara la edad del invididuo, compatibilizamos el formato en el que la tabla de equivalencias trata las edades con la forma en la que lo hace la EPH. Luego, hacemos lo mismo con el sexo de los encuestados creando una columna llamada "Genero". Una vez que ya podemos empezar a trabajar las dos bases conjuntamente, lo que primero hacemos (por cuestiones de poder de cómputo para realizar la siguiente tarea) es partir nuestra base en dos, una con respuestas sólo de mujeres y otras con sólo de varones. Creamos, en cada base, una lista vacía, y comenzamos un loop que busca en la base de la EPH y que contiene otro loop que busca en la base de la tabla de equivalencias, y que cuando coinciden en la columna "Edad" dos valores se suman a la lista vacía. Luego, creamos una nueva columna llamada "adulto_equiv" que nos otorga el valor de adulto equivalente que tiene una persona de tal edad y tal sexo. Finalmente, volvemos a unir las dos bases.
Para la segunda parte de este inciso agrupamos las observaciones por la columna "CODUSU" que permite juntar personas de un mismo hogar, y sumamos la cantidad de adultos equivalentes que hay en cada hogar y creamos un nuevo dataframe con esta nueva columna. Luego, unimos las dos bases para tener esta nueva columna en nuestra submuestra de la EPH.

1.3.

Para saber cuántos encuestados en nuestra muestra no reportaron sus ingresos contamos en cuántas observaciones aparece un 0 en la columna "ITF", el ingreso total familiar. Encontramos que 2904 personas no reportaron sus ingresos, más del 40 % del total.
Como por el resto de la sección nos interesa trabajar con quienes sí declararon sus ingresos totales familiares, partimos la muestra en 2, filtrando por si los valores en la columna "ITF" son mayores a 0, o iguales a 0. A los primeros los guardamos en una base llamada "respondieron" y a los segundos en "norespondieron".

1.4.

En este inciso le agregamos una nueva columna a la base "respondieron" llamada "ingreso_necesario" que nos dice cuál es el ingreso que necesita un hogar dado con cierta cantidad de adultos equivalentes para no ser pobre. Esta nueva columna la creamos con una función *lambda* a la cual a cada observación en la columna "ad_equiv_hogar" (que identifica la cantidad de adultos equivalentes en un hogar) se la multiplica por el ingreso necesario para que una persona no sea pobre, que en el Gran Buenos Aires es de $27.197,64[2].

[2]Como la consigna no especifica partir la base y luego necesitamos trabajar con el aglomerado de CABA también, asumimos que ese es el ingreso necesario para no ser considerado pobre también en CABA

1.5.

Una vez creada esta nueva columna definimos una función que denominamos *pobre* que nos devuelve un 1 si el ingreso total familiar es menor al ingreso necesario y un 0 si sucede lo contrario. Luego, creamos la columna "pobre" que surge de aplicarle la función *pobre* a nuestra base. Finalmente, para saber cuántos personas por debajo del ingreso necesario hay en nuestra muestra contamos la cantidad de veces que aparece un 1 en la columna "pobre". Así identificamos 1190 pobres en nuestra muestra, aproximadamente un 32 % del total.

2. Parte 2: Clasificación

2.1.

Para comenzar a trabajar en los metodos de clasificacion, borramos de las bases *respondieron* y *norespondieron*, creadas anteriormente, todas las variables relacionadas a los ingresos de las personas (ingresos per capita, ingresos familiares, ingresos por asalariados y trabajadores independientes, ocupación principales y otras, etc). Ademas, eliminamos las columnas relacionadas a las equivalencias de necesidades energéticas y al ingreso necesario por hogar para satisfacerlas.

2.2.

Como vimos en clase, el verdadero desafío de *machine learning* es minimizar el error de pronóstico *fuera* de la muestra. Por lo tanto, necesitamos partir a la muestra original en una muestra de entrenamiento y una de evaluacion. Para este ejercicio, definimos como muestra original a la base *respondieron* y la partimos de la siguiente forma:

- 70 % base de entrenamiento
- 30 % base de prueba

Luego, establecimos a la variable *pobre* como la variable dependiente, y al resto de las variables como independientes (matriz X). Sin embargo, consideramos que no todas las variables tenían que formar parte de las estimaciones, dado que algunas poseían gran cantidad de *missing values*. En particular, obviamos aquellas variables relacionadas con el tipo de ocupación (dado que la base contenía población desocupada/inactiva). Por lo tanto, en la matriz X únicamente dejamos las características de los miembros del hogar: 'CH06', 'CH04' , 'ESTADO', 'NIVEL_ED' , 'CH03' , 'CH07' , 'CH08', 'CH09' , 'CH10' , 'CH11', 'CH12' , 'CH13' , 'CH16','PP02C1', 'PP02C2', 'PP02C3', 'PP02C4', 'PP02C6', 'PP02E' y 'PP02H'[3].

2.3.

En esta subsección se explorarán distintos métodos de clasificación, con el sentido de generar un modelo que permite predecir si una persona es o no pobre en base a datos distintos al ingreso. Inicialmente, los modelos son entrenados con observaciones de individuos que sí reportaron sus ingresos, para luego poder utilizar el modelo con aquellos que no lo hicieron.

Logit

Lo que nos permite hacer el método logit es reemplazar la variable dependiente, los ingresos (que no conocemos), por su estimación mediante la regla de Bayes. Esto nos permite minimizar el riesgo eligiendo la opción que sea más probable para cada caso, en base a las variables explicativas.

En Python, esto lo logramos con el comando *LogisticRegression().* de la biblioteca scikit-learn, utilizando el método .fit() para crear el modelo con la base de entrenamiento, prediciendo luego los valores para la base de prueba, a través del método predict().

La matriz de confusión resultante es la del cuadro 1:

La curva ROC resultante es la de la figura 3

Finalmente, el valor de AUC (área bajo la curva ROC) es de 0,59 y la accuracy alcanza el 0,70.

[3]Estas últimas corresponden a maneras en las que el individuo estuvo buscando trabajo. No existen *missing values* ya que las personas con trabajo/que no buscan trabajo contestaban con un "cero"

		Y	
		0	1
$\widehat{Y}$	0	690	72
	1	265	97

Cuadro 1: Matriz de confusión logit

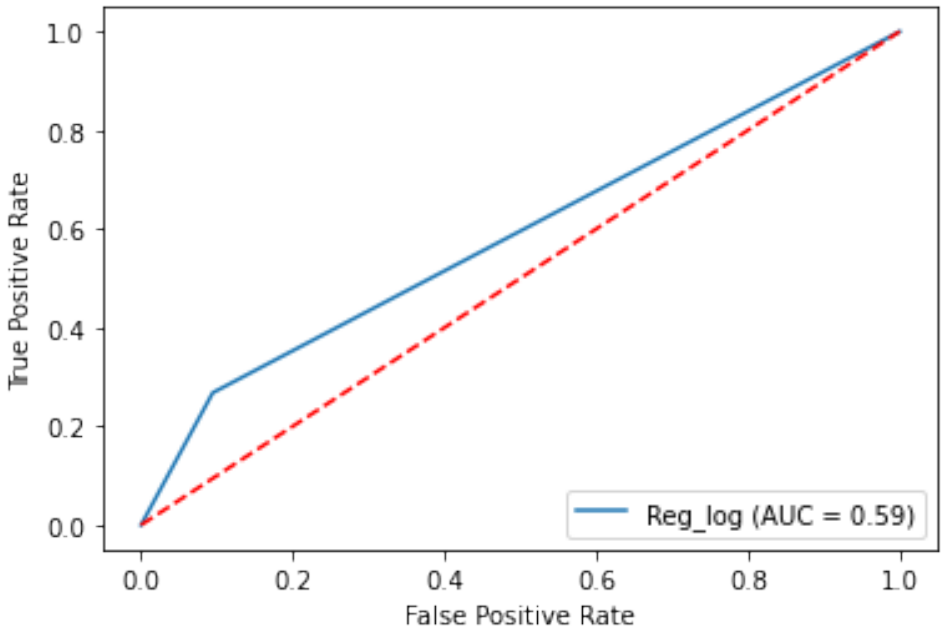


Figura 3: Curva ROC (Logit)

Análisis discriminante lineal

El análisis discriminante es otro método de clasificación que utiliza la regla de Bayes, estimando sus distintos componentes de forma separada para poder calcular las probabilidades relevantes para cada observación.

En Python, logramos formar un modelo de análisis discriminante con la función *LinearDiscriminantAnalysis()* de la biblioteca scikit-learn.

La matriz de confusión generada podemos verla en el cuadro 2.

		Y	
		0	1
$\widehat{Y}$	0	686	76
	1	271	91

Cuadro 2: Matriz de confusión análisis discriminante lineal

Por su parte, la curva ROC generada es la de la figura 4. El valor de AUC correspondiente es del 0,58, mientras que la precisión del modelo es del 0,69.

KNN

Por último, vamos a usar el método de K vecinos cercanos. Con un K=3, vamos a buscar las 3 observaciones más parecidas en cuanto al vector de variables explicativas para poder clasificar a una observación por fuera de la muestra. Es decir, en base a las características que observamos para los individuos, intentamos predecir cuales son pobres en base a sus 3 *vecinos* más cercanos.

En Python, conseguimos armar un modelo de KNN con la función *KNeighborsClassifier()* de la biblioteca scikit-learn.

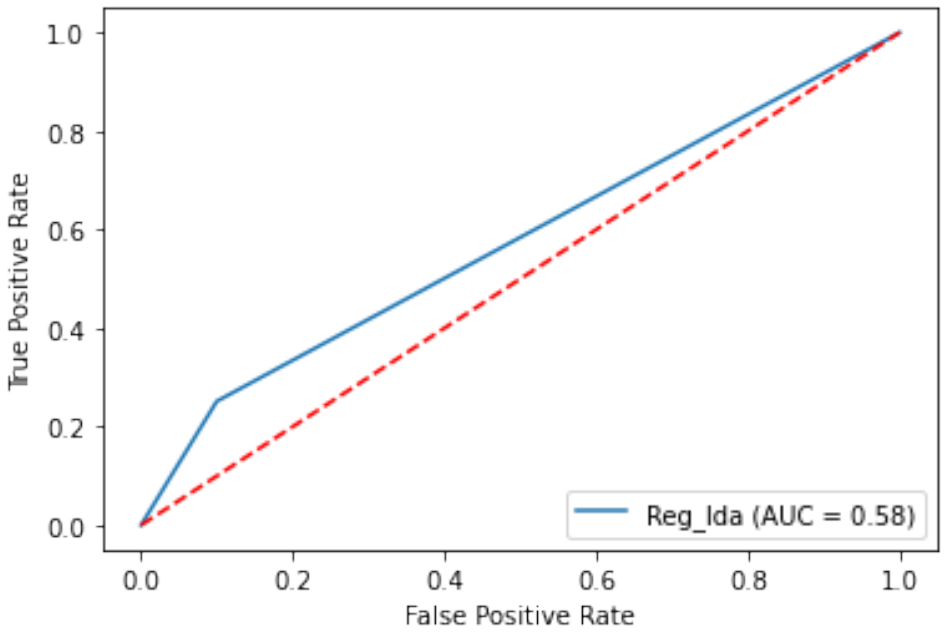


Figura 4: Curva ROC (LDA)

La matriz de confusión correspondiente a este método es la del cuadro 3.

La curva ROC del caso KNN la podemos ver en la figura 5. Finalmente, el valor de AUC es 0,64 y la accuracy del modelo alcanza el 0,71.

		Y	
		0	1
$\widehat{Y}$	0	636	126
	1	202	160

Cuadro 3: Matriz de confusión K vecinos cercanos

2.4.

Teniendo en cuenta los tres métodos, el que mejor predice es el de K vecinos cercanos.

En cuanto a **accuracy** supera a los otros dos métodos (0,71 contra 0,70 de logit y 0,69 de LDA) (definiendo a accuracy como el ratio $\frac{(TP+TN)}{P+N}$; la precisión para predecir verdaderos positivos y verdaderos positivos, en relación al total de positivos y negativos de la base).

El método de KNN también es superior en el valor arrojado de AUC (0,64, contra 0,59 de logit y 0,58 de LDA). Esta es el área bajo la curva ROC, valor que desearíamos que esté lo más cerca posible de 1. La intuición de esto es que la curva ROC ideal es aquella que alcanza una tasa total de verdaderos positivos sin tener ningún falso positivo, por lo que buscamos curvas que se parezcan lo más posible.

Por otro lado, si la idea de esta clasificación es poder identificar hogares pobres para incluirlos en alguna política de inclusión social, lo que nos va a interesar es maximizar la cantidad de *verdaderos positivos* ya que, suponiendo que tenemos cierta holgura en cuanto a la disponibilidad de recursos, lo mejor sería incluir la mayor cantidad de hogares que pueden ser considerados pobres. Es decir, nos afectaría menos darle fondos o ayuda a un hogar que no los necesita, que no incluir a alguien que sí lo necesite. Observando las matrices de confusión podemos ver que KNN alcanza 160 verdaderos positivos, contra 97 de logit y 91 de LDA.

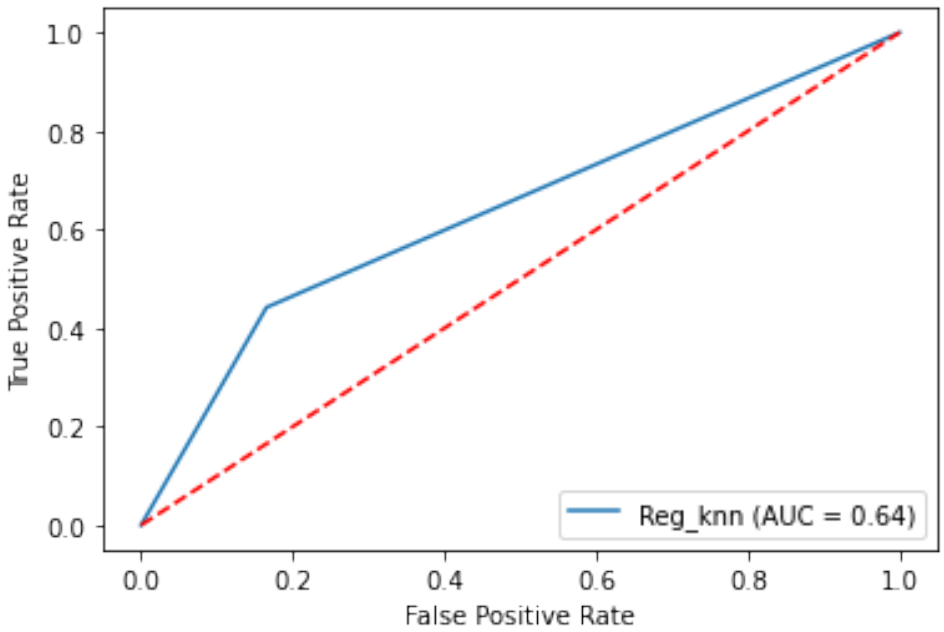


Figura 5: Curva ROC (KNN)

2.5.

Utilizando el modelo encontrado mediante el método de K vecinos cercanos para predecir con la base de los individuos que no respondieron datos de ingresos en la EPH, encontramos que la proporción de pobres en esta base alcanza el 29,3 %. Este dato es menor al que uno puede observar en los últimos datos del INDEC, por lo que podríamos suponer que existe un sesgo entre los individuos que no brindan sus datos de ingresos hacia estar más alejados de la línea de pobreza. Esto es consistente con los sesgos de selección identificados en la literatura para este tipo de encuestas.

2.6.

Creemos que para utilizar los distintos métodos de clasificación utilizamos algunas variables que podrían considerarse redundantes o poco correlacionadas con la predicción de la pobreza, algo que podría llevarnos a un problema de *overfit*.

Las variables explicativas elegidas para reducir las dimensiones del modelo logit son CH06 (edad, nos parece un buen predictor de la pobreza dada la marcada diferencia que hay entre los grupos etarios más jóvenes y los más viejos sobre la tasa de pobreza), ESTADO (la condición de actividad de la persona, algo que creemos clave para diferenciar los ingresos de personas adultas, dentro de un mismo grupo etario) y NIVEL_ED (nivel educativo de las personas, dato que nos puede agregar otra dimensión a la predicción de ingresos no laborales y sobre la estructura de los hogares).

Sin embargo, lo que encontramos realizando este experimento es que las medidas de precisión empeoran en la nueva regresión. El valor de verdaderos positivos pasa de 97 a 57, la *accuracy* desciende de 0,70 a 0,67 y el área bajo la curva ROC cae de 0,59 a 0,54.

Esto nos permite observar que el modelo anterior estaba mejor especificado, lejos del problema de overfit que podía preocuparnos. Es decir, quitar variables explicativas hizo que perdamos precisión en la clasificación, por lo que posiblemente estaban explicando factores relacionados a la pobreza que las variables de esta nueva regresión no llegan a alcanzar.

Referencias

Instituto Nacional de Estadísticas y Censos. 2016. La medición de la pobreza y la indigencia en Argentina.