

## Taller de Programación

---

### TRABAJO PRÁCTICO N° 3

#### CLASIFICANDO DE POBRES CON LA EPH

---

**Fecha de entrega:** 15 de noviembre a las 13:00 hs.

**Contenido:** Comenzar con la aplicación de los métodos de clasificación vistos en clase para identificar pobres usando la EPH y validación cruzada.

#### Modalidad de entrega

- El informe debe subirse a la carpeta correspondiente en repositorio de GitHub del grupo en formato PDF con el nombre **Program\_TP3\_Grupo#.pdf** (donde # es el número de grupo), incluyendo gráficos e imágenes dentro del mismo archivo. El mismo debe tener link al repositorio del grupo en la primer pagina. La extensión máxima es de **8 páginas (sin apéndices)** y se espera una redacción clara y precisa.
- Se debe publicar el código con los comandos utilizados en el repositorio, indicando claramente a qué inciso corresponde cada uno. El nombre del archivo deberá ser **Program \_TP3\_Grupo#**.
  - Al finalizar el trabajo práctico deben hacer un último commit en su repositorio de GitHub llamado “*Entrega final del TP*”.
  - El Jupyter Notebook y el correspondiente al TP3 deben estar dentro de esa carpeta.
  - La última versión en el repositorio es la que será evaluada. Por lo que es importante que:
    - No suban el pdf en la sección de “**Actividades/Entregas**” del campus hasta no haber terminado y estar seguros de que han hecho el *commit* y *push* a la versión final que quieren entregar.
    - No hagan nuevos *push* después de haber entregado su versión final. Esto generaría confusión acerca de que versión es la que quieren que se les corrija.
- Cualquier detección de copia o plagio será sancionada.

El objetivo de este trabajo práctico es intentar predecir si una persona es pobre o no utilizando distintas variables de características individuales que limpiaron y los distintos clasificadores vistos en clase. Recuerden que en los trabajos prácticos anteriores crearon dos bases de datos distintas: respondieron, que tiene datos de personas que sí respondieron su ingreso (ITF) y norespondieron, que tienen aquellas personas que no declaran su ingreso.

#### A. Enfoque de validación

Utilicen la base respondieron. Para cada año, dividan las observaciones en una base de prueba (test) y una de entrenamiento (train) utilizando el comando `train_test_split`. La base de entrenamiento debe comprender el 70% de los datos, y la semilla a utilizar (*random state instance*) debe ser 444. Establezca a pobre como su variable dependiente en la base de entrenamiento (vector *y*). El resto de las variables seleccionadas serán las variables independientes (matriz *X*). Recuerden agregar la columna de unos (1) para el intercepto. *Aclaración: no incluir la variable ingreso en X para predecir la pobreza porque cuando vayamos a la base de norespondieron no vamos a tener esa información.*

1. Cree una tabla de diferencia de medias entre la base de entrenamiento y la de testeo de las características seleccionadas en su matriz *X*. Para la matriz de las *X* seleccione variables que hayan limpiado en los TPs anteriores y justifique su inclusión para predecir la pobreza. Comente la tabla de la diferencia de medias de sus variables entre entrenamiento y testeo. ¿Hay diferencias significativas entre las medias del entrenamiento y testeo?
2. Separen la base respondieron en dos: `respondieron_2005` y `respondieron_2025`. Idem con la base norespondieron .

#### B. Modelo de Regresión Logística

3. Estimación y Efectos Marginales: Estimen una **Regresión Logística** usando `X_train` de `respondieron_2025`. Exporten una tabla con: (i) los coeficientes estimados para cada variable, (ii) los errores estándar y (ii) los odd-ratios. Interpreten los resultados de la tabla. (*Hint: en la clase 7 hay una ilustración de la tabla*).
4. Visualización: Grafiquen la  $\hat{P}(Y = 1|X)$  (en el eje vertical) y alguna característica **numérica** (en el eje horizontal). Comente dicho gráfico y la variable seleccionada para ilustrar la probabilidad de ser pobre según la característica seleccionada. (*Hint: en la clase 7 hay una ilustración de este estilo*).

### C. Método de Vecinos Cercanos (KNN)

5. Estimación: Clasifiquen a las observaciones como “pobre”/“no pobre” en su región con Vecinos Cercanos (**KNN**) usando  $K=\{1,5,10\}$  para su matriz  $X_{train}$  de respondieron\_2025. Expliquen en no más de 2-3 oraciones cómo la elección de K se relaciona con el trade-off sesgo varianza.
6. Visualización: Grafiquen dos características numéricas de su matriz  $X_{train}$  y visualicen las clases predichas por KNN usando con  $K=(1,10)$  con su frontera por clase “pobre”/“no pobre”.
7. K óptimo por Cross-validation: Dividan la base  $X_{train}$  de respondieron\_2025 en 5 partes (5-fold) para obtener el K óptimo por Cross-Validation con  $K=(1,10)$ . Llamenle a este modelo **KNN con K-CV**. Grafiquen el *accuracy* de cada modelo y comenten cual es el número óptimo de vecinos cercanos para identificar pobres.

### D. Desempeño de modelos fuera de la muestra, métricas y políticas públicas

8. Comparen el desempeño de predicción de pobreza en la base de  $X_{test}$  en 2025 de los modelos Logit y KNN con K-CV. Muestren:
  - i. la matriz de confusión (con umbral  $p > 0.5$ ) para logit,
  - ii. la curva ROC de ambos modelos en un solo gráfico,
  - iii. un tabla con dos métricas de clasificación vista en clase de ambos modelos, e interprete los resultados.
9. Suponga que el Ministerio de Capital Humano esta interesado en identificar a grupos vulnerables para dirigir los recursos de un programa de alimentos. Discutan cuál modelo de clasificación es “mejor” para predecir pobres y asignar dichos recursos de alimentos escasos. (*Hint*: recuerden que en la clase de clasificación discutimos el trade-off de minimizar error tipo I o II)
10. Con el método que seleccionaron, predigan qué personas son pobres dentro de la base norespondieron para 2025. ¿Qué proporción de las personas que no respondieron pudieron identificar como pobres?