Big Data - UNT

PRIMAVERA 2024

TRABAJO PRÁCTICO N 4

CLASIFICACIÓN Y REGULARIZACIÓN DE DESOCUPACIÓN USANDO LA EPH

Reglas de Formato y Presentación

Fecha de entrega: jueves 24 de octubre a las 13.59 hs.

Contenido: Análisis de hogares para los determinantes de la desocupacion, problema de clasificación de desempleo entre cohortes usando métodos de regularizacion y eleccion de hiperparámetros por cross-validation. Este trabajo, incluye la posibilidad para aquellos que quieran avanzar en las técnicas de programación en Python para sistemanizar el analisis de datos usando varios métodos vistos en clase.

Modalidad de entrega

Al finalizar el trabajo práctico deben hacer un último `commit` en su repositorio de GitHub con el mensaje Entrega final del TP.

- Asegúrense de haber creado una carpeta llamada TP4. Deben entregar un reporte (pdf) y el código (Jupyter notebook). Ambos deben estar dentro de esa carpeta.
- Deberán enviar el link a su repositorio -para que pueda ser clonado y corregido- al canal de Slack #tp-entregas
- La última versión en el repositorio es la que será evaluada. Por lo que es importante que:
    - No envíen el mensaje 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 trabajo. Esto generaría confusión acerca de qué versión es la que quieren que se les corrija.

Modalidad de entrega

- El informe debe ser entregado en formato PDF, con los gráficos e imágenes en este mismo archivo. Se espera una buena redacción en la resolución.

- Puede tener una extensión máxima de hasta 10 paginas (no se permite Apéndice).

- Entregar el código con los comandos utilizados, identificando claramente a que inciso corresponde cada comando.

- **Importante**: Todos los miembros del equipo deben haber hecho al menos un commit durante la realización del TP para asegurar que todos hayan aportado a su resolución.

Parte I: Análisis de la base de hogares y tipo de ocupación

Ahora que ya están familiarizados con la Encuesta Permanente de Hogares (EPH) y la desocupación, vamos a complejizar un poco la construcción de las tasas del desempleo. Relacionaremos la información a nivel hogar.

1. Exploren el diseño de registro de la base de hogar: a priori, ¿qué variables creen pueden ser predictivas de la desocupación y seria útil incluir para perfeccionar el ejercicio del TP3? Mencionen estas variables y justifiquen su elección.

2. Descarguen la base de microdatos de la EPH correspondiente al primer trimestre de **2004** y **2024** en formato .dta y .xls, respectivamente. La base de hogares se llama `Hogar_t104.dta` y `usu_hogar_T124.xls`, respectivamente. Eliminen todas las observaciones que **no** corresponden a los aglomerados de Gran Tucumán - Tafí Viejo y unan ambos trimestres en una sola base. Unan, a la base de la encuesta individual de cada año, la base de la encuesta de hogar. **Asegúrese de estar usando las variables CODUSU y NRO_Hogar para el merge.**

3. Limpien la base de datos tomando criterios que hagan sentido. Explicar cualquier decisión como el tratamiento de valores faltantes (*missing values*), extremos (*outliers*), o variables categóricas. Justifique sus decisiones.

4. Construya variables (mínimo 3) que no estén en la base pero que sean relevantes para predecir individuos desocupados (por ejemplo, la proporción de personas que trabajan en el hogar).

5. Presenten estadísticas descriptivas de cinco variables de la encuesta de hogar que ustedes creen que pueden ser relevantes para predecir desocupación. Comenten las estadísticas obtenidas.

6. En el TP3 calcularon la tasa de desocupación según INDEC y economía laboral, para el 1er trimestre de 2024. Utilice una sola observación por hogar y sumen el ponderador PONDERA que permite expandir la muestra de la EPH al total de la población que representa ¿Cuál es la tasa de hogares con desocupación para Tucumán? ¿se asemeja dicha tasa a la reportada en el INDEC en sus informes?

Parte II: Clasificación y regularización

El objetivo de esta parte del trabajo es nuevamente intentar predecir si una persona está desocupada o no. Esta vez utilizando distintas variables de características individuales y preguntas de la encuesta de hogar. A su vez incluiremos ejercicios de regularización y de validación cruzada.

1. Para cada año, partan la base respondieron en una base de prueba y una de entrenamiento (`X_train, y_train, X_test, y_test`) 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 101. Establezca a desocupado como su variable dependiente en la base de entrenamiento (vector y). El resto de las variables serán las variables independientes (matriz X). Recuerden agregar la columna de unos (1).
2. Expliquen como elegirían λ por validación cruzada. Detallen por qué no usarían el conjunto de prueba (test) para su elección.
3. En validación cruzada, ¿cuáles son las implicancias de usar un k muy pequeño o uno muy grande? Cuando $k = n$ (con n el número de muestras), ¿cuántas veces se estima el modelo?
4. Implementen la penalidad, L1 como la de LASSO y L2 como la de Ridge, para regresión logística usando la opción `penalty` y reporten la matriz de confusión, la curva ROC, los valores de AUC y de Accuracy para cada año.[1] ¿Cómo cambiaron los resultados con respecto al TP3? ¿Mejor o empeoró la performance de regresión logística con regularización?
5. Realicen un barrido en $\lambda = 10^n$ con $n \in \{-5, -4, -3 \ldots, +4, +5\}$ y utilicen 10-fold CV para elegir el λ óptimo en regresión logística con Ridge y con LASSO. ¿Qué λ seleccionó en cada caso? Usando la librería de seaborn, generen box plot mostrando la distribución del error de predicción para cada λ. Cada box debe corresponder a un valor de λ y contener como observaciones el error medio de validación para cada partición. Además, para la regularización LASSO, generen un line plot, pero ahora graficando el promedio de la proporción de variables ignoradas por el modelo en función de λ, es decir la proporción de variables para las cuales el coeficiente asociado es cero.[2]

[1] En la clase 8, vimos el método de regularización en regresión lineal donde la variable dependiente es numérica. En este caso, nuestra variable dependiente es binaria (ocupado, desocupado), por lo que usamos la regresión logística y aprovechamos la opción de penalidad para aplicar los métodos de regularización visto en clase.

[2] *Hint*: a mayor penalidad, esperamos que más coeficientes sean 0, por lo tanto, esta figura debe tener una forma de "S".

6. En el caso del valor óptimo de λ para LASSO encontrado en el inciso anterior, ¿qué variables fueron descartadas? ¿Son las que hubieran esperado? ¿Tiene relación con lo que respondieron en el inciso 1 de la Parte I?
7. Elijan alguno de los modelos de regresión logística donde hayan probado distintos parámetros de regularización y comenten: Compare los resultados de 2004 versus 2024, ¿qué método de regularización funcionó mejor: Ridge o LASSO? Comenten mencionando el error cuadrático medio (ECM).

Parte III: Construcción de funciones y sistematización de métodos (Opcional)

El objetivo de esta parte del trabajo es generar código que sea flexible y que este modularizado (en funciones bien documentadas con `docstrings`). De esta forma, evitarán repetir código y podrán utilizarlo en distintos escenarios (como por ejemplo la Parte II 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 la matriz de confusión, las curvas ROC y los valores de AUC y de accuracy score de cada método). El output de la función debe ser una colección con las métricas evaluadas.

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, solo X e y). Pueden ayudarse con la función KFold para generar las particiones necesarias.

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` obtengan el error promedio para cada configuración.[3] Finalmente, la función debe devolver la configuración que genere menor error.[4]
4. Escriban una función llamada `evalua_multiples_metodos` que les permita implementar los siguientes métodos con los hiperparámetros que ustedes elijan. Para la regresión logística, asegúrense de que esta

[3] Utilicen la medición del error que prefieran. Una opción sería el Error Cuadrático Medio (MSE).

[4] Consejo: cuanto mas genérica construyan la función, luego podrá ser utilizada en mas situaciones. Por ahora, la usaremos solo para buscar el λ optimo cuando utilicemos regularización.

función utilice su función `evalua_config` para optimizar el λ de la regularización. 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:[5]

- Regresión logística
- Análisis de discriminante lineal
- KNN

[5] *Hint 2*: para la regresion logistica, cuando incluyan regularizacion observen que deberan correr la funcion `evalua_metodo` dos veces. Una para optimizar los hiperparametros (con un set de datos para train y otro para validacion) y otra para obtener las metricas con el hiperparametro optimo (con un set de datos para train y otro para test).