



Clasificación y validación cruzada

Objetivo del Trabajo Práctico 02

Evaluar lo visto en clase sobre clasificación y selección de modelos, utilizando validación cruzada.

Enunciado

En el presente TP trabajaremos con el conjunto de datos de imágenes denominado **TMNIST**¹. Cada imagen del set de datos representa un dígito representado en una determinada tipografía. En el link ubicado a pie de página pueden acceder a una descripción más detallada del dataset.

Para comenzar deben **descargar del campus de la materia** el conjunto de datos, el cual se encuentra en formato csv.

Fecha de entrega: **12 de Noviembre de 2024, 23:50hs.** Al igual que el TP-01, la entrega de este TP se realizará a través del campus de la materia.

Ejercicios

1. Realizar un **análisis exploratorio** de los datos. Entre otras cosas, deben analizar la cantidad de datos, cantidad y tipos de atributos, cantidad de clases de la variable de interés (dígitos) y otras características que consideren relevantes. Además se espera que con su análisis puedan responder las siguientes preguntas:
 - a. ¿Cuáles parecen ser atributos relevantes para predecir el número al que corresponde la imagen? ¿Cuáles no? ¿Creen que se pueden descartar atributos?
 - b. ¿Hay números que son más parecidos entre sí? Por ejemplo, ¿Qué es más fácil de diferenciar: las imágenes correspondientes al dígito 1 de las de el dígito 3, ó las del dígito 3 del dígito 8?
 - c. Tomen una de las clases, por ejemplo el dígito 0, ¿Son todas las imágenes muy similares entre sí?
 - d. Este dataset está compuesto por imágenes, esto plantea una diferencia frente a los datos que utilizamos en las clases (por ejemplo,

¹ **TMNIST**. <https://www.kaggle.com/datasets/nimishmagre/tmnist-typeface-mnist>

el dataset de Titanic). ¿Creen que esto complica la exploración de los datos?

Importante: las respuestas correspondientes a los puntos 1.a, 1.b y 1.c deben ser justificadas en base a gráficos de distinto tipo.

Ayuda: Para ayudarles en la representación gráfica les dejamos un poco de código para orientarles.

```
#####  
# Plot imagen  
img = np.array(X.iloc[12]).reshape((28,28))  
plt.imshow(img, cmap='gray')  
plt.show()  
#####
```

2. **(Clasificación binaria)** Dada una imagen se desea responder la siguiente pregunta: **¿la imagen corresponde al dígito 0 o al dígito 1?**

- A partir del dataframe original, construir un nuevo dataframe que contenga sólo al subconjunto de imágenes correspondientes a los dígitos 0 o 1. Sobre este subconjunto de datos, analizar cuántas muestras se tienen y determinar si está balanceado con respecto a las dos clases a predecir (la imagen es del dígito 1 o del dígito 0).
- Separar los datos en conjuntos de train y test.
- Ajustar un modelo de KNN en los datos de train, considerando pocos atributos, por ejemplo 3. Probar con distintos conjuntos de 3 atributos y comparar resultados. Analizar utilizando otras cantidades de atributos. Para comparar los resultados de cada modelo usar el conjunto de test generado en el punto anterior.

OBS: Utilicen métricas para problemas de clasificación como por ejemplo, exactitud.

- Comparar modelos de KNN utilizando distintos atributos y distintos valores de k (vecinos). Para el análisis de los resultados, tener en cuenta las medidas de evaluación (por ejemplo, la exactitud) y la cantidad de atributos.

Observación: en este ejercicio 2 no estamos usando k-folding ni estamos dejando un conjunto held-out. Solamente entrenamos en train y evaluamos en test, donde train y test están fijos a lo largo de los incisos c,d,e.

3. **(Clasificación multiclase)** Dada una imagen se desea responder la siguiente pregunta: **¿A cuál de los 10 dígitos corresponde la imagen?**

- Separar el conjunto de datos en desarrollo (dev) y validación (held-out).
Para los incisos b y c, utilizar el conjunto de datos de desarrollo. Dejar apartado el conjunto held-out en estos incisos.
- Ajustar un modelo de árbol de decisión. Probar con distintas profundidades (entre 1 y 10).

- c. Realizar un experimento para comparar y seleccionar distintos árboles de decisión, con distintos hiperparámetros. Nuevamente, limitarse a usar profundidades entre 1 y 10.
Para esto, utilizar validación cruzada con k-folding. ¿Cuál fue el mejor modelo? Documentar cuál configuración de hiperparámetros es la mejor, y qué performance tiene.
- d. Entrenar el modelo elegido a partir del inciso previo, ahora en todo el conjunto de desarrollo. Utilizarlo para predecir las clases en el conjunto held-out y reportar la performance.
OBS: Al realizar la evaluación utilizar métricas de clasificación multiclase como por ejemplo la exactitud. Además pueden realizar una matriz de confusión y evaluar los distintos tipos de errores para las clases.

Grupos

Los grupos deben estar conformados por 3 (y sólo 3) integrantes. Ni más, ni menos. Deberán i) registrar la conformación del grupo en la siguiente planilla, y ii) definir quién va a ser el encargado del envío (debe ser uno y sólo uno de los integrantes del grupo):

https://docs.google.com/spreadsheets/d/1X8pzZ9v-PuH_n7FA8PIQ_N5yvsSGRpjseXioc2xhHzs/edit?usp=sharing

NOTA: Si mantienen la conformación del grupo tal cual la del TP01, mantengan el MISMO nombre.

Acerca de la entrega

Para la entrega deberán preparar los siguientes archivos:

- Un archivo llamado *tmnist_nombregrupo.py* con el código principal. Este archivo puede complementarse con otros archivos .py donde figure parte del código, y que sean importados y utilizados desde el archivo principal.

Como siempre, ordenar el código de la siguiente manera:

- Al inicio, una descripción que contemple: el nombre del grupo, los nombres de los participantes, contenido del archivo y cualquier otro dato relevante que considere importante.
- Luego la sección de los imports.
- A continuación, la carga de datos.
- Siguiendo, las funciones propias que hayan definido.
- Y finalmente, el código que no está dentro de funciones.

El código debe estar modularizado (separando bloques con `#%%`) para permitir su ejecución por fragmentos.

Todo lo que figure en el informe debe deducirse de los resultados del código.

Importante: Incluir un archivo README.txt con los requerimientos de bibliotecas utilizadas e instrucciones de cómo ejecutar el código.

- Un informe breve (no más de 10 carillas) en pdf llamado *informe_tp2_nombregrupo.pdf*. Además deben entregar una copia impresa.

Ordenar el informe de la siguiente manera:

- Breve introducción al problema donde se muestre el análisis exploratorio realizado.
- Explicación sobre los experimentos realizados, incluyendo los gráficos que consideren convenientes.
- Conclusiones, incluyendo los resultados relevantes de los modelos desarrollados.

Importante: ¡No deben entregar los archivos del dataset!

Autoevaluación

Al finalizar la entrega, y **antes de enviar el TP-02**, realizar lo siguiente:

- a. **Copiar** la siguiente planilla de autoevaluación (una sola a nivel grupal) **a una carpeta personal**:

<https://docs.google.com/spreadsheets/d/1rdOa4W8U816WhARikGY9mL8f9SBAbTBV5bE2lyCI8mk/edit?usp=sharing>

- b. Completarla
- c. Descargarla como pdf y agregarla al envío virtual y en papel.