



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 **Sign Language MNIST**¹. Cada imagen del set de datos representa una letra en lenguaje de señas americano. 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: **10 de Marzo 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 (letras en lenguaje de señas) 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 la letra a la que corresponde la seña? ¿Cuáles no? ¿Creen que se pueden descartar atributos?
 - b. ¿Hay señas que son parecidas entre sí? Por ejemplo, ¿Qué es más fácil de diferenciar: la seña de la E de la seña de la L o la seña de la E de la seña de la M?
 - c. Tomen una de las clases, por ejemplo la seña correspondiente a la C, ¿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 **Sign Language MNIST**. <https://www.kaggle.com/datasets/datamunge/sign-language-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.

2. Dada una imagen se desea responder la siguiente pregunta: **¿la imagen corresponde a una señal de la L o a una señal de la A?**

- A partir del dataframe original, construir un nuevo dataframe que contenga sólo al subconjunto de imágenes correspondientes a señales de las letras L o A.
- Sobre este subconjunto de datos, analizar cuántas muestras se tienen y determinar si está balanceado con respecto a las dos clases a predecir (la señal es de la letra L o de la letra A).
- Separar los datos en conjuntos de train y test.
- Ajustar un modelo de KNN considerando pocos atributos, por ejemplo 3. Probar con distintos conjuntos de 3 atributos y comparar resultados. Analizar utilizando otras cantidades de atributos.

Importante: Para evaluar 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.

3. **(Clasificación multiclase)** Dada una imagen se desea responder la siguiente pregunta: **¿A cuál de las vocales corresponde la señal en la imagen?**

- Vamos a trabajar con los datos correspondientes a las 5 vocales. Primero filtrar solo los datos correspondientes a esas letras. Luego, separar el conjunto de datos en train y test.
- Ajustar un modelo de árbol de decisión. Analizar distintas profundidades.
- Para comparar y seleccionar los árboles de decisión, utilizar validación cruzada con k-folding.

Importante: Para hacer k-folding utilizar los datos del conjunto de train.

- ¿Cuál fue el mejor modelo? Evaluar cada uno de los modelos utilizando el conjunto de test. Reportar su mejor modelo en el informe.
- OBS: Al realizar la evaluación utilizar métricas de clasificación multiclase. Además pueden realizar una matriz de confusión y evaluar los distintos tipos de errores para las clases.

Acerca de la entrega

Para la entrega deberán preparar los siguientes archivos:

- Un archivo llamado *sign_nombreyapellido.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 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_nombreyapellido.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!