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 **Kuzushiji-MNIST**[1] [1] Cada imagen del set de datos representa un carácter japonés antiguo manuscrito de entre 10 tipos distintos. Se trata de un dataset que busca ser similar a otro muy famoso llamado MNIST[2].
En el link ubicado a pie de página pueden acceder a una descripción más detallada del dataset. Ahí pueden ver los nombres de cada clase.
Para comenzar deben **descargar del campus de la materia** el conjunto de datos, el cual se encuentra en formato csv. Encontrarán 2 archivos. En uno tienen el mapeo de cada hiragana antiguo escrito en estilo kuzushi (caracteres japoneses cursivos antiguos) con su clase y en el otro el mapeo de cada clase con la representación en píxeles de la imágen correspondiente.

Fecha límite para la entrega: Domingo 09 de noviembre de 2025, 23:50hs. Al igual que el TP-01, la entrega de este TP se realizará a través del campus de la materia. El día **lunes 10 de noviembre, antes de las 12:30**, **deben entregar el informe impreso junto con la autoevaluación** en secretaría del Departamento de Computación, 1er piso (Sala 2105). La Secretaría del DC está abierta desde las 9. Sólo si así se los indican en secretaría, entreguen o en Planta baja (Sala 1502, Atención a estudiantes), abierta desde las 12.

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 (caracteres japoneses) 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 tipo de carácter al que corresponde la imagen? ¿Cuáles no? ¿Creen que se pueden descartar atributos?

[1] **Kuzushiji-MNIST** https://github.com/rois-codh/kmnist
[2] **MNIST** https://en.wikipedia.org/wiki/MNIST_database

b. ¿Hay caracteres que son más parecidos entre sí? Por ejemplo, ¿Qué es más fácil de diferenciar: las imágenes correspondientes a la clase 2 de las de la clase 1, ó las de la clase 2 de la clase 6?
c. Tomar una de las clases, por ejemplo la clase 8, ¿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, 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 código para orientarlos.

```
############
# 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 a la clase 4 o a la clase 5?**
   a. A partir del dataframe original, construir un nuevo dataframe que contenga sólo al subconjunto de imágenes correspondientes a las clases 4 y 5. Sobre este subconjunto de datos, analizar cuántas muestras se tienen y determinar si está balanceado con respecto a las dos clases a predecir (si la imagen es de la clase 4 o de la clase 5).
   b. Separar los datos en conjuntos de train y test.
   c. Ajustar un modelo de KNN sobre los datos de entrenamiento utilizando una cantidad reducida de atributos (por ejemplo, 3). Probar con distintos conjuntos de atributos seleccionados a partir del análisis exploratorio -por ejemplo, varios subconjuntos distintos de 3 atributos si se eligió ese número- y comparar los resultados obtenidos. Repetir el análisis utilizando diferentes cantidades de atributos. Para comparar los resultados de cada modelo usar el conjunto de test generado en el punto anterior.
      OBS: Utilizar métricas de evaluación para problemas de clasificación. Para ello, elijan las métricas de evaluación que consideren más adecuadas y expliquen por qué decidieron utilizarlas.
   d. Comparar modelos de KNN utilizando distintos atributos y distintos valores de k (vecinos). Para el análisis de los resultados, tener en cuenta las métricas de evaluación utilizadas y la cantidad de atributos utilizados.

Observación: en este Ejercicio 2 no estamos usando k-folding. Solamente entrenamos en train y evaluamos en test, donde train y test están fijos a lo largo de los incisos c y d.

3. **(*Clasificación multiclase)*** Dada una imagen se desea responder la siguiente pregunta: **¿A cuál de las 10 clases corresponde la imagen?**
    a. 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.
    b. 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 hiperparámetos. 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.

    Observación: 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): *TP-02-Grupos*

NOTA: Si mantienen la conformación del grupo tal cual la del TP-01, por favor mantengan el MISMO nombre.

Acerca de la entrega
Para la entrega deberán preparar los siguientes archivos:

1. Un archivo llamado *TP-02-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, un encabezado con una descripción que contemple: el nombre del grupo, los nombres de lxs 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.

2. Un archivo llamado README.txt con los requerimientos de bibliotecas utilizadas e instrucciones de cómo ejecutar el código.
3. Un informe breve (no más de 10 carillas) en pdf llamado *TP-02-Informe-nombregrupo.pdf* . Además deben entregar una copia impresa (el jueves próximo siguiente a la entrega virtual).

   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.
   - Bibliografía. Citen el paper [1].
4. La planilla de autoevaluación que se explica a continuación.

Importante: No deben entregar los archivos del dataset.

Autoevaluación

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

1. **Copiar** la siguiente planilla de autoevaluación (una sola a nivel grupal) **a una carpeta personal**:
   TP-02-Autoevaluación
2. Completarla
3. Descargarla como pdf y agregarla al envío virtual y en papel.

Referencias

[1] Clanuwat, Tarin, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. "Deep learning for classical japanese literature." *arXiv preprint arXiv:1812.01718* (2018).