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 de caracteres tipeados. Cada imagen del set de datos representa una letra mayúscula del alfabeto en inglés (sin la ñ) tipeada con diversas tipografías.
Para comenzar deben **descargar del campus de la materia** el conjunto de datos, el cual se encuentra en formato csv y que fue obtenido a partir del dataset *English Typed Alphabets and Numbers*.

Al igual que el TP-01, la entrega de este TP se realizará de manera virtual a través del campus de la materia, así como presencial en papel.

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 (la letra representada) 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 imagen? ¿Cuáles no? ¿Creen que se pueden descartar atributos?
    b. ¿Hay letras que son más parecidas entre sí? Por ejemplo, ¿Qué es más fácil de diferenciar: las imágenes correspondientes a la letra S de las de la letra M, ó las de la letra O de la letra Q?
    c. Tomar una de las letras, por ejemplo la letra J, ¿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 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 a la letra O o a la letra L?**
   a. A partir del dataframe original, construir un nuevo dataframe que contenga sólo al subconjunto de imágenes correspondientes a las letras O y L. 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 letra O o de la letra L).
   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 la exactitud como métrica para problemas de clasificación.
   d. 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.

3. ***(Clasificación multiclase)*** Dada una imagen se desea responder la siguiente pregunta: ***¿A cuál de las 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 20).
    c. Realizar un experimento para comparar y seleccionar distintos árboles de decisión, con distintos hiperparámetos. 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 letras del conjunto held-out y reportar la performance.

    Observación: Al realizar la evaluación utilizar la exactitud como métrica de clasificación multiclase. 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):

2026V 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.

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:

a. **Copiar** la siguiente planilla de autoevaluación (una sola a nivel grupal) **a una carpeta personal**:
   2026V TP-02-Autoevaluacion
b. Completarla
c. Descargarla como pdf y agregarla al envío virtual y en papel.