

Trabajo Práctico 2

El objetivo del presente trabajo práctico es aplicar algunas herramientas de lo aprendido sobre clasificación y selección de modelos con validación cruzada.

El trabajo práctico se realiza en grupos de 3 integrantes, sin excepción. Se asume que los grupos serán los mismos que en el TP1. Si por alguna razón necesitan cambiar la composición del grupo, por favor avisen a los docentes.

Enunciado

Dataset. En el presente TP trabajaremos con el conjunto de datos de imágenes denominado **Fashion-MNIST** [1]. Cada imagen del set de datos tiene 28x28 píxeles en escala de grises con valores 0-255 y representa una prenda de ropa entre 10 tipos distintos. Se trata de un dataset de estructura similar a otro muy famoso llamado **MNIST** [2]. En la sección Referencias tienen links para acceder a una descripción más detallada del dataset, como por ejemplo la organización de las columnas y los nombres de cada clase. Para comenzar deben [descargar del campus de la materia](#) el archivo con el conjunto de datos que se encuentra en formato csv. Ahí también encontrarán un breve script de Python para levantar los datos y graficar una imagen.

Clasificación por atributos. Este dataset está compuesto por imágenes, lo que plantea una diferencia frente a los datos que utilizamos en las clases. En `titanic`, por ejemplo, cada elemento del dataset estaba definido por atributos con una interpretación muy concreta (sexo, edad, etc). En este trabajo, en cambio, los atributos son el valor de cada píxel en la imagen ($28 \times 28 = 784$ atributos). Tengan en cuenta esto al realizar la exploración de los datos y utilicen gráficos para ayudar a la visualización!

Entrega. Fecha límite para la entrega: **Lunes 16 de Junio de 2025, 23:50hs**. Al igual que el TP1, la entrega del TP2 se realizará a través del campus de la materia.

Ejercicios

1. **Análisis exploratorio.** Entre otras cosas, deben analizar la cantidad de datos, cantidad y tipos de atributos, cantidad de clases de la variable de interés (prendas de ropa) y otras características que consideren relevantes. Además se espera que con su análisis puedan responder las siguientes preguntas:
 - a. ¿Cuáles parecen ser atributos (i.e. píxeles) relevantes para predecir el tipo de prenda al que corresponde la imagen? ¿Cuáles no? ¿Creen que se pueden descartar atributos?
 - b. ¿Hay prendas que son más parecidas entre sí? Por ejemplo, ¿qué es más fácil de diferenciar: las imágenes correspondientes a la clase 2 de las de la clase 1, o 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?

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

2. **Clasificación binaria.** Dada una imagen “incógnita” se desea responder la siguiente pregunta: ¿la imagen corresponde a la clase 0 o a la clase 8?

- A partir del dataframe original, construir un nuevo dataframe que contenga sólo al subconjunto de imágenes correspondientes a las clases 0 y 8. 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 0 o de la clase 8).
- Separar los datos en conjuntos de training y testing.
- Ajustar un modelo de kNN sobre los datos de entrenamiento utilizando una cantidad reducida de atributos (por ejemplo, 3). Probar con distintos subconjuntos 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.

Importante: Utilizar 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 no estamos usando k-folding ni estamos dejando un conjunto held-out. Solamente entrenamos con train y evaluamos con test, donde train y test están fijos a lo largo de los incisos c-d.

3. **Clasificación multiclase.** Dada una imagen “incógnita” se desea responder la siguiente pregunta: ¿a cuál de las 10 clases 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. **No utilizar** el conjunto held-out en estos incisos.
- Ajustar un modelo de árbol de decisión. Probar con distintas profundidades máximas (entre 1 y 10).
- Realizar un experimento para comparar y seleccionar distintos árboles de decisión, con distintos hiperparámetros. Nuevamente, limitarse a usar profundidades máximas 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.
- 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.

Acerca de la entrega

Importante: ¡No deben entregar los archivos del dataset!

La entrega comprende los siguientes CUATRO archivos:

1. Un archivo llamado `TP-Grupo_XX.py` con el código principal (XX = número de grupo). 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/número del grupo, los nombres de lxs participantes, contenido del archivo y cualquier otro dato relevante que consideren importante.
- Luego la sección de los imports.
- A continuación, la carga de datos.
- Siguiendo, las funciones propias que hayan definido.
- 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 `TP2-Grupo_XX-Informe.pdf`.


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.

Autoevaluación

Al finalizar la entrega, y **antes de enviar el TP2**, realizar lo siguiente:

- a. **Copiar** la siguiente planilla de autoevaluación (una sola a nivel grupal) **a una carpeta personal**:  `TP2-Autoevaluacion`



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

Tomen esto como una herramienta: si ven que pueden mejorar el puntaje en algún ítem de la autoevaluación, obviamente pueden rehacer o mejorar la parte correspondiente antes de entregar.

Referencias

[1] Fashion-MNIST: <https://github.com/zalandoresearch/fashion-mnist>

[2] MNIST: https://en.wikipedia.org/wiki/MNIST_database