TALLER DE PROCESAMIENTO DE SEÑALES

2do Cuatrimestre 2026 - Trabajo Práctico Nº 4 - LDA, QDA y KNN

Se desea predecir el nivel de confort térmico de una oficina. Para ello se cuenta con la base de datos *ASHRAE Global Thermal Comfort Database*.

(**a**) *Explotación de datos:*

1. Descargar los archivos `https://github.com/CenterForTheBuiltEnvironment/ashrae-db-II/raw/master/v2.1.0/db_metadata.csv` y `https://github.com/CenterForTheBuiltEnvironment/ashrae-db-II/raw/master/v2.1.0/db_measurements_v2.1.0.csv.gz`.
2. Leer la documentación del repositorio.

(**b**) *Pre-procesamiento:*

1. *Construcción del dataset:*
   - Completar los datos de temperatura exterior faltante, con el reporte de la estacion meteorológica. Posteriormente, eliminar las columnas asociadas a la estación.
   - Agregar a cada registro los datos del edificio.
   - Eliminar todas las columnas de identificación (id).
   - Eliminar las columnas de preferencia subjetiva, exceptuando `thermal_preference` (será utilizada como nuestra etiqueta a predecir).
   - Eliminar los las columnas con índices de confort termico (por ejemplo pmv) ya que estos son funciones de otros datos.
   - Eliminar todas las columnas que tengan mas de un 50 % de datos faltantes.
2. *Codificación:*
   - Dentro de los predictores, distiguir entre variables numéricas y categóricas.
   - Dentro de los predictores categóricos, distiguir entre ordinales y nominales. Codificar adecuadamente estas variables. ☞: Recordar que si hay solo dos clases se codifica como ordinal.
3. *Completar datos faltantes:* Se desea implementar un *imputador*. Cada dato faltante debe ser completado a partir del siguiente procedimiento:
   - Para cada valor 'Nan' (faltante), de una muestra, descartamos todos los otros features que sean 'Nan'.
   - Filtramos como vecinos potenciales, aquellos que no sean 'Nan' en el feature a completar o ningun otro feature sobreviviente.
   - Buscamos los K vecinos más cercanos entre los vecinos potenciales.
   - Completamos el dato faltante con la media de los vecinos cercanos.

   ☞: Puede ser útil heredar primitivas de sklearn como `BaseEstimator`, `TransformerMixin` y `ClassifierMixin`, para habilitar el uso de `Pipeline` y `ColumnTransformer`. Ejemplo:
   class CustomKNNImputer(BaseEstimator, TransformerMixin):.

(**c**) *Clasificador*:

1. Separar el conjunto de datos en *entrenamiento* y *testeo* con las proporciones 75 % y 25 %.
2. Implementar LDA.
3. Implementar QDA.
4. Reportar *accuracy*, *macro-f1* y *matriz de confusión* tanto para entrenamiento como testeo.

☞: Puede ocurrir que la matriz de covarianza a invertir no tenga rango completo. En cuyo caso se deberá regularizar como $\tilde{\Sigma} = \Sigma + \lambda \cdot I$.

(**d**) *Generación de muestras sintéticas:*

1. Para cada etiqueta, se desea simular 10 muestras.
2. Para cada etiqueta, modificar el modelo generativo anterior para simular solamente algunos predictores (a partir del resto). Simular las características personales de cada individuo a partir de las variables relacionadas con el ambiente y el edificio. Justificar el procedimiento. ☞: Si los predictores se dividen en $X = [X_1, X_2]$, simulamos $X_1|_{Y=y, X_2=x_2}$. Se recomienda utilizar las expresiones de los momentos siguiendo el *Matrix Cookbook*.
3. Analice cualitativamente las muestras generadas. Considere imponer restricciones a las muestras generadas utilizando síntesis con rechazo.