Práctica 2. Extensión del sistema de clasificación de datos

Propósito: Comprender las ventajas que ofrece un buen diseño de un sistema software para el mantenimiento y extensión de la funcionalidad del mismo.

Enunciado: Se trata de extender la funcionalidad de la aplicación software desarrollada en la práctica 1 sobre "Clasificación de datos utilizando el método *k*-nn". Se pide incorporar al programa las siguientes *mejoras*:

1.- Mostrar **información** relevante sobre un **dataset** cargado: número de atributos (*p* columnas), nombre de los atributos, número de casos (*n* filas).

2.- Mostrar **información** relevante sobre los **atributos**: Para los cuantitativos: nombre, mínimo valor, máximo valor, media ($\bar{x} = \sum_{i=1}^{n} {}^{x_i}/_{n}$), desviación típica ($\sigma = \sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2/n}$. Para los cualitativos: nombre, número de clases distintas, valores y frecuencias relativas de cada clases.

3.- Opciones de **preprocesado de datos**: sin preprocesado (datos crudos), rango 0-1 ($\hat{x}_i = \frac{x_i - mi_{\ i}}{max_i - mi_{\ i}}$) y estandarización ($\hat{x}_i = \frac{x_i - \overline{x}}{\sigma}$). El defecto es rango 0-1.

4.- Pesado de atributos: Consiste en asignar diferentes pesos a los atributos predictivos del modelo (según su importancia), lo cual influye en el cálculo de las distancias. Por defecto, todas las variables predictivas tienen igual peso (1).

5.- Configuración del algoritmo de clasificación:

Establecer el valor de *k*.

El usuario puede **seleccionar** diferentes **métricas** para calcular las distancias entre unas instancias y otras:

Distancia Euclidea: Es el defecto.

$$d(x, \quad y) = \sqrt{\sum_{i=1}^{p}(x_i - y_i)^2}$$

Distancia Manhattan: $$d(x, \quad y) = \sum_{i=1}^{p}|x_i - y_i|$$

Distancia de Chebychef: $$d(x, \quad y) = \max_{i=1..p}|x_i - y_i|$$

Permitir las siguientes variantes adicionales del método *k*-nn a la hora de determinar la clase predicha:

Pesado de casos:
Consiste en asignar a cada caso (instancia) entre los *k* vecinos más cercanos un peso (en principio, diferente) según diversos criterios:

- Igualdad de votos: Los *k* vecinos más cercanos tienen igual peso (1). Es el defecto para el pesado de casos. Todos los casos pesan lo mismo.
- Cercanía: El voto que aporta un caso es inversamente proporcional a la distancia a la que se encuentra de la instancia a clasificar.
- Voto fijo según el orden de vecindad: El usuario especifica pesos fijos decrecientes a los vecinos primero, …, *k*-ésimo, suponiendo que se encuentran ordenados de menor a mayor distancia respecto al caso a clasificar. El defecto para esta opción es $w_1 = k$, …, $w_k = 1$.

Regla de clasificación:
El usuario puede especificar la regla de clasificación:

- La clase más votada (*mayoría simple*). Es el defecto para la regla.
- La clase más votada cuyo número de votos supere un cierto umbral de votos. Cuando el umbral es del 50% la regla se denomina *mayoría absoluta*. El usuario debe introducir el umbral (decimal de escala 2 entre 0 y 1). Si no lo hace, el defecto es 0.50 (mayoría absoluta).

6.- Mostrar **información** sobre los diferentes parámetros de **configuración del algoritmo** de clasificación (valor de k, métrica usada, pesado de casos y regla de clasificación utilizada).

7.- Modo de experimentación:

- Permitir **generar** automáticamente, a partir de un dataset cargado, los **conjuntos de entrenamiento y pruebas**. El usuario especificará qué porcentaje del dataset se dejará para el conjunto de pruebas. El complementario será el porcentaje del conjunto de entrenamiento. Los conjuntos de entrenamiento y pruebas son disjuntos.
- El usuario indicará si quiere que la generación del conjunto de pruebas sea aleatoria o no. Si no es aleatoria, las instancias del conjunto de pruebas serán las últimas del dataset. Si es aleatoria, el usuario introducirá una semilla (por defecto, seed = 1234) y la distribución a usar será uniforme sin repetición en el rango de filas del dataset.
- Permitir guardar en el disco los conjuntos de entrenamiento y prueba (para poder replicar los experimentos).
- Permitir leer conjuntos de entrenamiento y pruebas previamente creados para realizar experimentos.
- Resultados del experimento: Usando las instancias ya clasificadas del conjunto de pruebas, calcular y mostrar la **precisión predictiva** (número de instancias del conjunto de pruebas clasificadas correctamente dividido por el número de instancias totales del conjunto de pruebas), así como la **matriz de confusión** (matriz de datos que informa sobre el número de instancias del conjunto de pruebas que perteneciendo a una clase determinada fueron mal clasificadas en otra - predicciones incorrectas).