PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

Tarea 2

IIC2433 – Minería de Datos

Fecha de Entrega: Jueves 10 de Octubre hasta las 23:59 hrs

En esta tarea pondremos en práctica los algoritmos de clasificación vistos en clases. Para ello usaremos una base de datos astronómicos disponible en el sitio web del curso en la carpeta "Tarea 2". La primera fila indica los nombres de cada variable. La primera columna corresponde al Id de cada objeto (Macho_id), la última columna corresponde a la clase del objeto, las demás columnas corresponde al resto de las variables. Los nombres relativos a cada número de clase son los siguientes: None Variable: 2, Quasar: 3, Be Stars: 4, Cepheid: 5, RR Lyrae: 6, Eclipsing Binaries: 7, MicroLensing: 8, Long Periodic Variable: 9. Apoyándonos en el toolbox de python scikit learn (http://scikit-learn.org/) deben realizar los siguientes pasos:

- Obtener Recall, Precisión y F-Score para cada clase, usando 10-Fold cross validation, para los clasificadores: K-NN ($K = \{5, 8, 15\}$). Genere una tabla comparativa de los indicadores para los distintos valores de K.
- Obtener Recall, Precisión y F-Score para cada clase, usando 10-Fold cross validation, para un árbol de decisión. Cada nodo debe realizar splits binarios, es decir, preguntar si la variable es mayor o menor igual que cierto valor. Dicho valor debe ser determinado en forma automática, simplemente como el valor que genera una mayor ganancia de información para ese atributo. Esto viene implementado en la clase "sklearn.tree.DecisionTreeClassifier ".
- Obtener Recall, Precisión y F-Score para cada clase, usando 10-Fold cross validation, para un clasificador Naive Bayes. Muestre para cada atributo un histograma de cómo se distribuyen los elementos de cada clase a lo largo del atributo (discretize en 20 bins). Use los nombres de las clases en la leyenda (no los números que representan las clases).
- Generar una tabla comparando los resultados de cada clasificador.

La tarea debe ser entregada en un archivo comprimido que contenga: el código Python, un .pdf con el informe de resultados y un archivo Readme.txt con las instrucciones para ejecutar el código. El código debe estar debidamente comentado. Se habilitará un buzón en el sitio del curso cerca de la fecha de entrega para que puedan subir su archivo, **no se aceptan entregas atrasadas**, la tarea es **individual**.