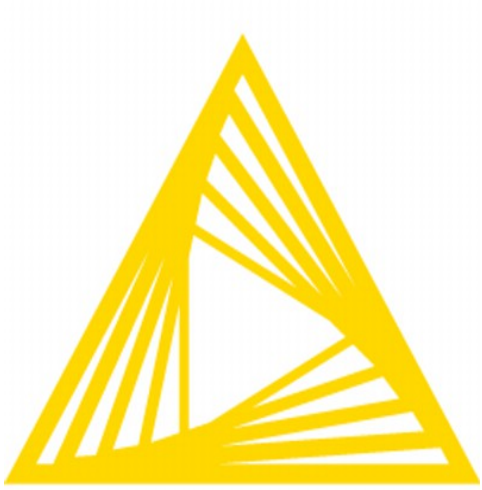
UNIVERSIDAD DE GRANADA
E.T.S. de Ingenierías Informática y de Telecomunicación

Departamento de Ciencias de la
Computación e Inteligencia Artificial

Inteligencia de Negocio



PRÁCTICA 2

COMPETICIÓN EN KAGGLE SOBRE PREPROCESADO DE DATOS

Christian Andrades Molina

ÍNDICE

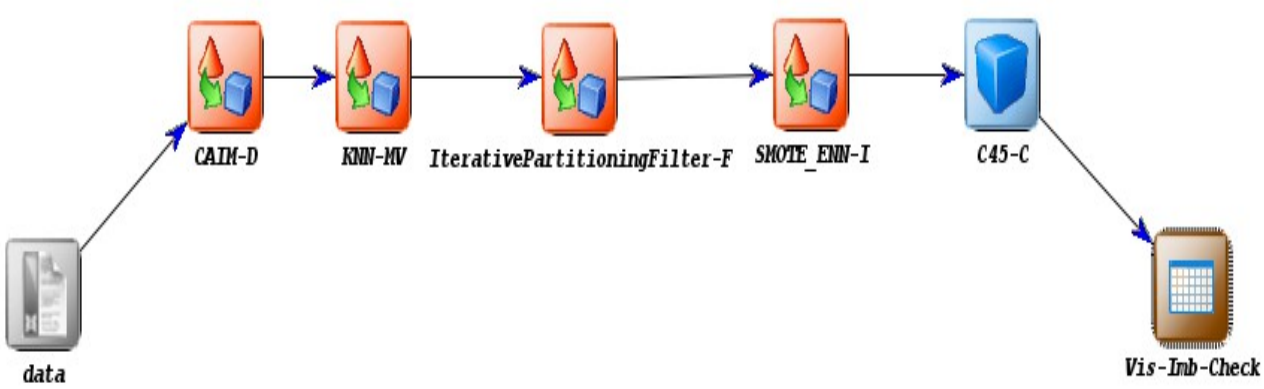
1. Problema a resolver

La base de datos para esta práctica se basa en una base mayor utilizada en bioinformática para predecir estructuras de proteinas. Está compuesta por un conjunto de instancias clasificadas por positivas o negativas, con una tasa de instancias de clase positiva del 25%.

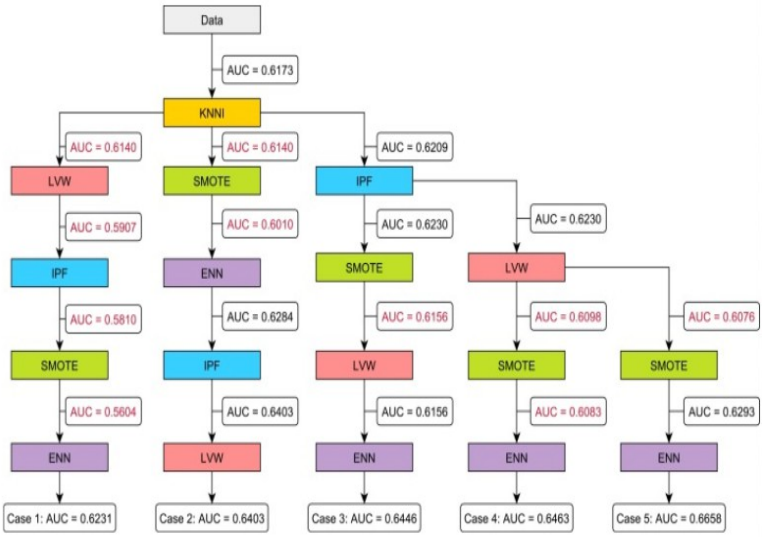
Tenemos un conjunto de datos de entrenamiento con 50 características y 20000 instancias. El algoritmo a utilizar par clasificar será C4.5. Para evaluar las soluciones, se emplea el valor del área bajo la curva ROC.

2. Solución final

Esta secuencia de nodos es la solución final para mi práctica, con un 0.71828 tras varias pruebas con diferentes métodos sin éxito. Para ello, he cambiado el orden de la primera solución aplicando en primer lugar una reducción de datos con el nodo CAIM-D, transformando los valores continuos/discretos en nominales y a partir de esa modificación de los datos, facilitar la tarea procesando los valores perdidos y filtrar el ruido, consiguiendo un mejor resultado:



Para ello he utilizado de orientación un diagrama de las diapositivas ofrecidas por la asignatura:

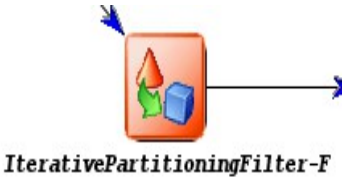


La secuencia de pasos utilizada es esta:

Nodo con el dataset utilizado para la práctica con todas las operaciones ejecutadas siguiendo el guión instrucciones_keel.txt

Reducción de datos: llevamos a cabo una reducción de los datos una vez tratado el ruido y los valores perdidos mediante la técnica de discretización, transformando los valores continuos/discretos en nominales.

Valores perdidos: este nodo pertenece a uno de los clasificados para tratar los valores perdidos, utilizando las instancias más cercanas.



Filtrado de datos con ruido: este nodo pertenece a uno de los 3 filtros de ruido más conocidos. En este caso, elimina datos ruidosos en múltiples iteraciones hasta una condición de parada.

SMOTE-I (Synthetic Minority Over-sampling Technique) es un algoritmo que para los ejemplos de la clase más minoritaria (25% la clase positiva) introduce ejemplos sintéticos donde en la unión entre el elemento seleccionado y sus vecinos más cercanos. La amplicación usada (**SMOTE-ENN-I**), este método utiliza como siguiente paso los vecinos más cercanos para filtrar objetos ruidosos

3. Soluciones implementadas

Partimos de una solución eficiente tras varias pruebas con resultados bajos y a partir de ella intentamos mejorarlo hasta la solución final:

1. data – ipfilter-f – KNN-MV – CAIM-D – SMOTE_ENN-I – C45-C – Vis-Imb-Check – Score: 0.69680
2. data – ipfilter-f – KNN-MV – Chi2-D– SMOTE_I – C45-C – Vis-Imb-Check – Score: 0.61131
3. data – ipfilter-f – KNN-MV – C45-C – Vis-Imb-Check – Score: 0.66156
4. data – ipfilter-f – KNN-MV – CAIM-D – SMOTE_I – C45-C – Vis-Imb-Check – Score: 0.62122
5. data – ipfilter-f – KNN-MV – SMOTE_-I – C45-C – Vis-Imb-Check – Score: 0.61608
6 .data – ipfilter-f – KNN-MV – SMOTE-I - ipfilter-f – C45-C – Vis-Imb-Check – Score : 0.64705
7 .data – ipfilter-f – KNN-MV – CACC-D – SMOTE_RSB-I – C45-C – Vis-Imb-Check - ERROR
8. data – ipfilter-f – KNN-MV – CACC-D – SMOTE_TL-I – C45-C – Vis-Imb-Check - Score: 0.69143
9. data – ipfilter-f – KNN-MV – CADD-D – SMOTE_RSB-I – C45-C – Vis-Imb-Check - ERROR
10 . data – ipfilter-f – KNN-MV – CACC-D –Borderline_SMOTE-I– C45-C – Vis-Imb-Check -Score: 0.67267
11. data – CAIM-D - KNN-MV – pfilter-f –SMOTE_ENN-I – C45-C – Vis-Imb-Check – Score: 0.71828
12. data – CAIM-D - pfilter-f - KNN-MV – pfilter-f –SMOTE_ENN-I – C45-C – Vis-Imb-Check – Score: 0.70885