MINERÍA
PRÁCTICA 01

PABLO SIMÓN SAINZ

IVÁN RUIZ GÁZQUEZ

ÍNDICE

En esta segunda práctica haremos un estudio del modelo KNN (k nearest neighbours).

El dataset sobre el que trabajaremos recoge varios datos médicos, y los emplearemos para hacer una predicción sobre si el paciente padece una enfermedad cardiovascular.

CONJUNTO DE DATOS

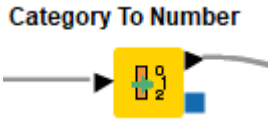
Dado que el algoritmo se basa su entrenamiento en las distancias sobre el campo de búsqueda, es importante modificar los datos para poder hacer este cálculo correctamente.

PRIMERA FASE: ADAPTACIÓN PARA DISTANCIAS

En este caso, hemos identificado tres escenarios en la primera fase de procesamiento de datos:

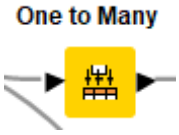
- **Valores numéricos**, para estos no tendremos que realizar cambios de primera mano.
- **Valores categóricos**:
    - **Con distancias internas**, en este caso nos encontramos valores como 'AgeCategory', que nos encontramos con grupos de edad los cuales cada categoría tiene una distancia diferente a otra (un rango de edad pequeño estará más cerca de uno mediano que de uno grande).

      En este dataset los de este tipo son solamente lineales, por lo que bastará con usar el nodo 'Category To Number', para dar valores del 0 a n.

      Un problema que nos hemos encontrado es que hemos tenido que gestionar a parte la asignación de números debido a que KNIME asigna el número en función del orden en el que se va encontrando una categoría nueva.

    - **Sin distancias internas**, valores como la Raza o los de "YES"/"NO", en un principio no tienen variaciones de distancia entre ellas, por lo que aplicaremos 'One shoot' mediante el nodo 'One to Many'.

SEGUNDA FASE: NORMALIZACIÓN Y AJUSTE DE POBLACIÓN

En este caso también aplicaremos normalización a los datos como hicimos en la práctica anterior.

En cuanto a la población, nos encontramos con que existen más pacientes sin problemas del corazón que con, por lo que esta está un tanto desbalanceada.

Esto provocará una mejor adaptación en una clase que en otra, por lo que nos interesa o bien duplicar elementos de la minoritaria, o bien eliminar elementos de la mayoritaria.

ACC SIN SMOTE

KNIME

File Hilite

HeartDisea...	No	Yes
No	2829	94
Yes	237	37

Correct classified: 2.866 Wrong classified: 331

Accuracy: 89,647% Error: 10,353%

Cohen's kappa (κ): 0,135%

PYTHON

File Hilite

HeartDisea...	No	Yes
No	2842	81
Yes	236	38

Correct classified: 2.880 Wrong classified: 317

Accuracy: 90,084% Error: 9,916%

Cohen's kappa (κ): 0,149%

El problema del oversampling es que, aunque nos aumente el accuracy, no es un accuracy real, ya que estamos duplicando los datos de entrada (test y entrenamiento). Si incluyéramos un nuevo dataset de test, este tendría menor accuracy.

EJEMPLO DE OVERSAMPLING X2

Un ejemplo de oversampling de x2 nos otorga el siguiente accuracy con KNIME:

File Hilite

HeartDisea...	No	Yes
No	8724	45
Yes	73	749

Correct classified: 9.473 Wrong classified: 118

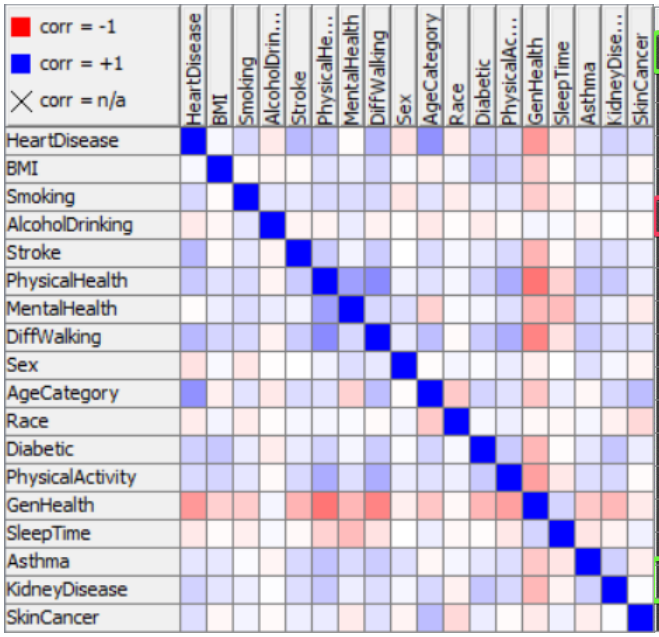
Accuracy: 98,77% Error: 1,23%

Cohen's kappa (κ): 0,92%

El accuracy es de 98,77. Pero es un accuracy que se adapta mucho a los datos insertados, y que no se adaptaría correctamente a nuevos datos.

CORRELACIÓN

En comparación con la práctica anterior, encontramos correlaciones relativamente bajas, por lo que obtendremos la clasificación mediante varios parámetros con un coeficiente medio-bajo, con un rango de [0.012, 0.431].



Row ID	HeartDisease
GenHealth	-0.403371208320095
Sex	-0.11308688181937702
SleepTime	-0.08498194523433142
AlcoholDrinking	-0.08243839104262024
Race	-0.07479762844753164
MentalHealth	-0.01291970295412069
BMI	0.028081861457949147
Asthma	0.09728977457800567
SkinCancer	0.1265302097860466
PhysicalActivity	0.14351723531061322
Smoking	0.1534309198547714
KidneyDisease	0.17645916349282809
Diabetic	0.1840971340899298
PhysicalHealth	0.21067335924860156
Stroke	0.272938610171068
DiffWalking	0.2774665141974658
AgeCategory	0.43056206867382396
HeartDisease	1.0

En este caso hemos obtenido peores resultados al realizar el filtro, por lo que en este caso no lo aplicaremos para los experimentos.

KNIME

File Hilite

HeartDisea...	No	Yes
No	2372	551
Yes	47	2876

Correct classified: 5.248 Wrong classified: 598

Accuracy: 89,771% Error: 10,229%

Cohen's kappa (κ): 0,795%

PYTHON

File Hilite

HeartDisea...	No	Yes
No	2369	554
Yes	48	2875

Correct classified: 5.244 Wrong classified: 602

Accuracy: 89,702% Error: 10,298%

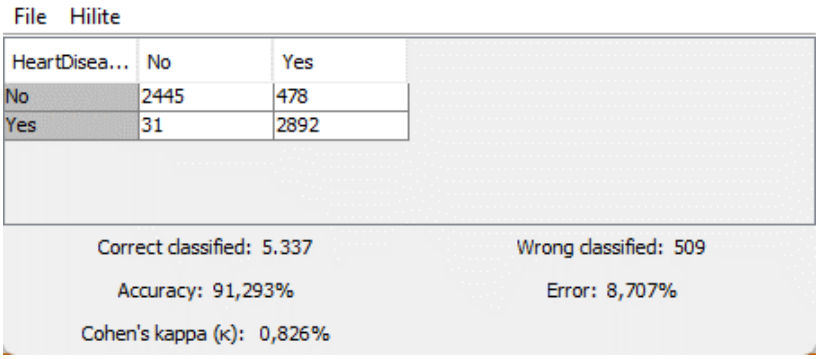
Cohen's kappa (κ): 0,794%

EXPERIMENTOS

A continuación, haremos comparaciones a distintas variaciones en el modelo para poder analizar las diferencias y ver cuál es el método más adecuado y por qué.

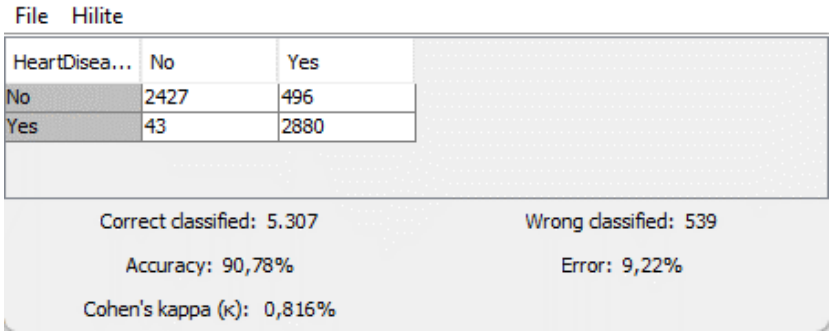
KNIME MODEL

Como modelo pivote utilizaremos el creado mediante funciones de KNIME, ya que nos ha resultado más sencillo de manipular.



PYTHON MODEL

Comparando este modelo respecto a KNIME, tenemos un resultado muy parecido, tanto en accuracy como en kappa. Solo que al parecer uno clasifica mejor los de una clase que el otro.



K NEIGHBOURS COMPARACIÓN

5 VECINOS

KNIME

File Hilite

HeartDisea...	No	Yes
No	2306	617
Yes	42	2881

Correct classified: 5.187 Wrong classified: 659

Accuracy: 88,727% Error: 11,273%

Cohen's kappa (κ): 0,775%

PYTHON

File Hilite

HeartDisea...	Yes	No
Yes	2881	42
No	619	2304

Correct classified: 5.185 Wrong classified: 661

Accuracy: 88,693% Error: 11,307%

Cohen's kappa (κ): 0,774%

7 VECINOS

KNIME

File Hilite

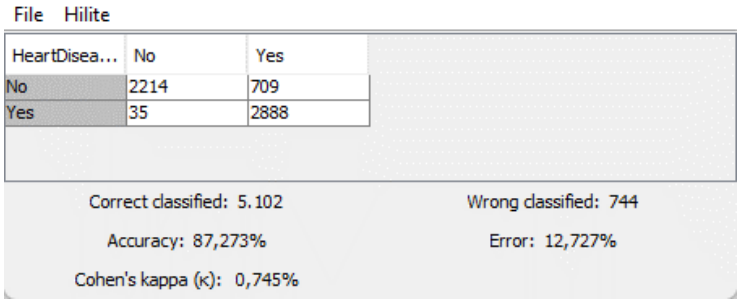
HeartDisea...	No	Yes
No	2207	716
Yes	36	2887

Correct classified: 5.094 Wrong classified: 752

Accuracy: 87,137% Error: 12,863%

Cohen's kappa (κ): 0,743%

PYTHON


File Hilite
HeartDisea... No Yes
No 2214 709
Yes 35 2888
Correct classified: 5.102
Wrong classified: 744
Accuracy: 87,273%
Error: 12,727%
Cohen's kappa (κ): 0,745%

DIFERENCIA ENTRE FUNCIONES DE DISTANCIA

KNIME

DISTANCIA EUCLÍDEA:

File Hilite

HeartDisea...	No	Yes
No	2440	483
Yes	34	2889

Correct classified: 5.329 Wrong classified: 517

Accuracy: 91,156% Error: 8,844%

Cohen's kappa (κ): 0,823%

DISTANCIA MANHATTAN:

File Hilite

HeartDisea...	No	Yes
No	2445	478
Yes	31	2892

Correct classified: 5.337 Wrong classified: 509

Accuracy: 91,293% Error: 8,707%

Cohen's kappa (κ): 0,826%

PYTHON

DISTANCIA EUCLÍDEA:

File Hilite

HeartDisea...	No	Yes
No	2393	530
Yes	35	2888

Correct classified: 5.281 Wrong classified: 565

Accuracy: 90,335% Error: 9,665%

Cohen's kappa (κ): 0,807%

DISTANCIA MANHATTAN:

File Hilite

HeartDisea...	No	Yes
No	2427	496
Yes	43	2880

Correct classified: 5.307 Wrong classified: 539

Accuracy: 90,78% Error: 9,22%

Cohen's kappa (κ): 0,816%

CONCLUSIÓN

Obviando las conclusiones de la práctica anterior.

Respecto al tratamiento de datos, se ha añadido dificultad con los datos nominales y sus distintos tratamientos para el cálculo de distancias. Y en cuanto a la cantidad de población hemos comprobado que para que no pierda peso y accuracy para la clase minoritaria, debemos ajustar esta cantidad para equilibrar e incluso mejorar el rendimiento general.

Por la otra parte, en el ajuste del modelo, hemos visto que en el KNN en este caso funciona mejor con distancia manhattan, lo más probable que sea debido a que da mayor peso a las distancias de los valores nominales.

Como dato interesante, hemos notado a la herramienta KNIME le toma más tiempo realizar operaciones con nodos de Python. Esto puede ser debido al paso de lenguajes.