MINERÍA
PRÁCTICA 01

PABLO SIMÓN SAINZ

IVÁN RUIZ GÁZQUEZ

ÍNDICE

INTRODUCCIÓN

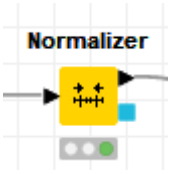
Para esta práctica haremos una comparación entre algunos modelos de clasificación disponibles en la herramienta KNIME.

TRATAMIENTO DEL DATASET

En este caso, el dataset se nos presenta en tres ficheros csv a los que tendremos que hacer ciertos tratamientos previos al entrenamiento del modelo de clasificación.

NORMALIZACIÓN

Nos encontramos con unos datos con valores con una diferencia significativa, por lo que es interesante normalizarlos para que todos cuenten.



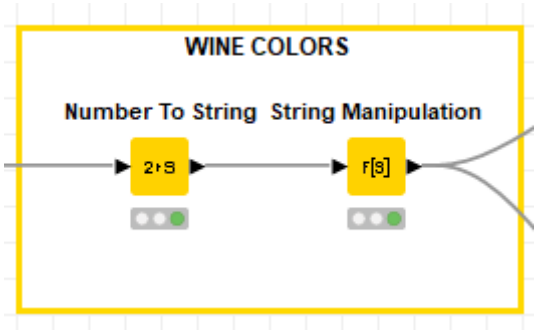
1. Nodo de normalización

Tenemos la suerte de que todos los datos de entrada son todos numéricos por lo que solamente tendremos que buscar que estos valores se encuentren entre 1 y 0.

Para ello se emplea la fórmula:

$$d_n = (d - d_{min}) / (d_{max} - d_{min})$$

En cuanto al dato de entrada puede ser interesante renombrar los datos para su posterior análisis, ya que en este caso los datos son {1, 2, 3}.



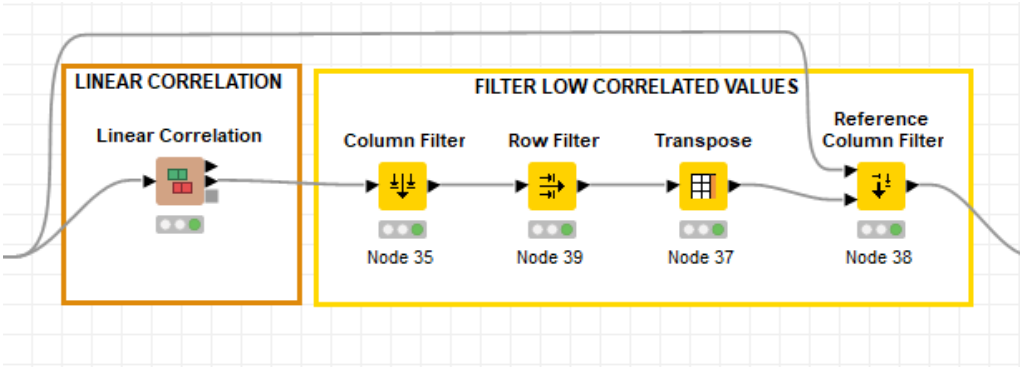
2. Nodos de traducción de números a string. Pasamos de valores (1-3) a "Blanco, Tinto y Claro"

CORRELACIÓN

Los dataset cuentan con mucha información útil con la que entrenaremos nuestro modelo de clasificación.

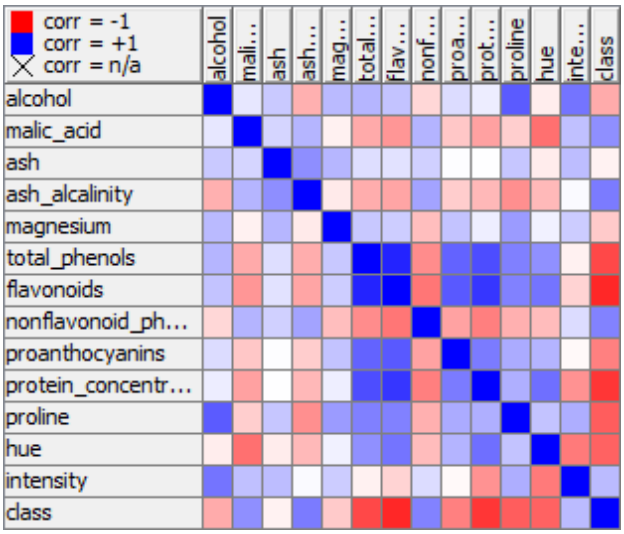
Sin embargo, a la hora de obtener los datos, podemos toparnos con que algunos atributos empleados no son tan importantes para la clasificación. Por ello es interesante limpiar algunos datos para aumentar la eficiencia del modelo.

Para poder evitar esta situación utilizaremos el coeficiente de correlación, para poder comprobar qué parámetros son más prescindibles.



3. Estudio de correlación y filtro de valores poco relacionados con la clase. En nuestro caso el atributo "ash" es el menos correlacionado con la clase

Echando un vistazo a la matriz de correlación, en la fila de la 'clase', vemos que atributos como 'ash', 'magnesium' o 'intensity' no guardan mucha correlación con la clase.

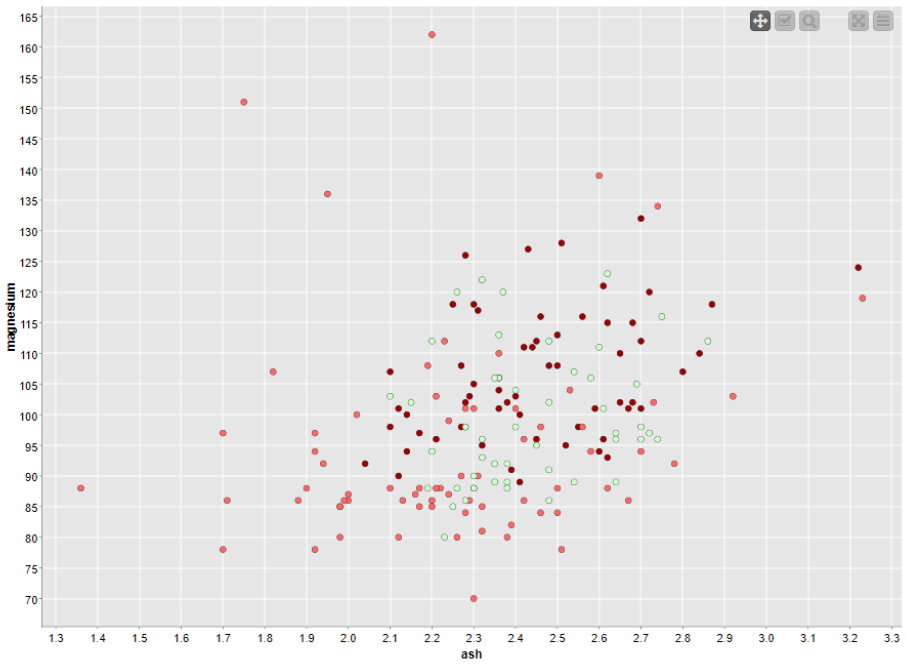


4. Matriz de correlación. En blanco los atributos que menos se correlacionan entre sí (valor neutro 0)

Como hemos fijado el límite en *0.1*, 'ash' queda fuera de la operación.

EJEMPLO CORRELACIÓN BAJA

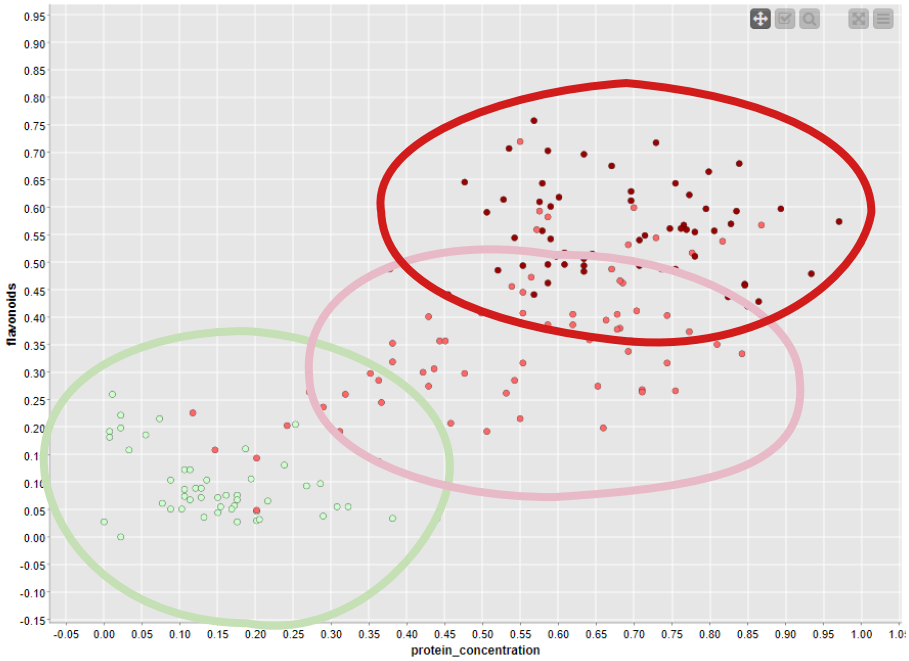
El “ash” y el “magnesium” no tienen mucha correlación con la clase como acabamos de comentar arriba.



5. Diagrama de dispersión de las clases en la correlación entre "ash" y "magnesium", los dos valores que menos aportan en la clasificación. No vemos las clases separadas de forma clara.

EJEMPLO CORRELACIÓN ALTA

Flavonoids con protein_concentraton mantienen una correlación alta con la clase.

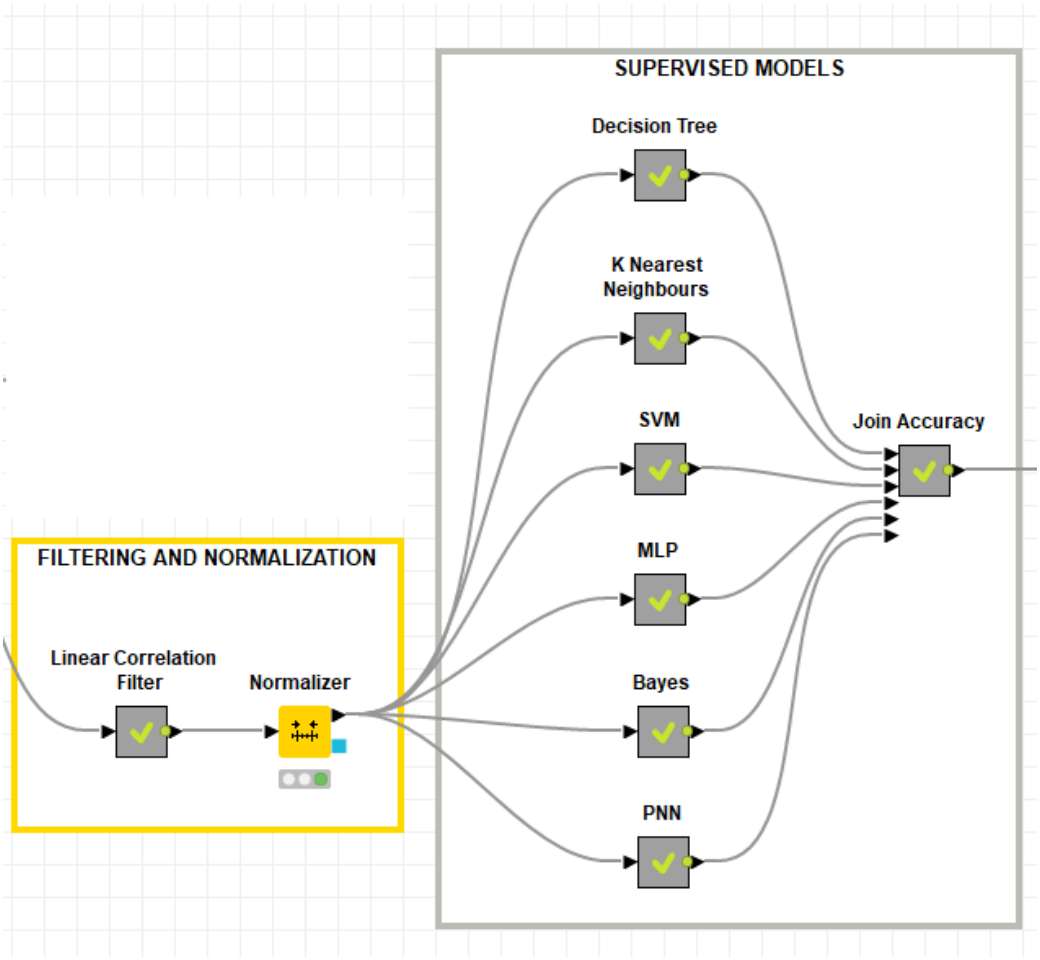


6. Diagrama de dispersión de las clases en la correlación de "protein_concentration" y "flavonoids". Estos dos atributos son los que más aportan a la clase. (Ver imagen 4). Podemos observar la diferencia clara entre los distintos tipos de vino.

MÉTODOS DE CLASIFICACIÓN

A parte del árbol de decisión, hemos probado 5 modelos más.

- K Vecinos más cercanos.
- SVM (Máquina de soporte vectorial).
- MLP (Perceptrón Multicapa).
- Clasificador de Bayes.
- PNN. Red neuronal probabilística.



7. Estructura de nodos de filtro y normalización (explicados en los apartados anteriores), junto a la lista de modelos a entrenar con la salida unida.

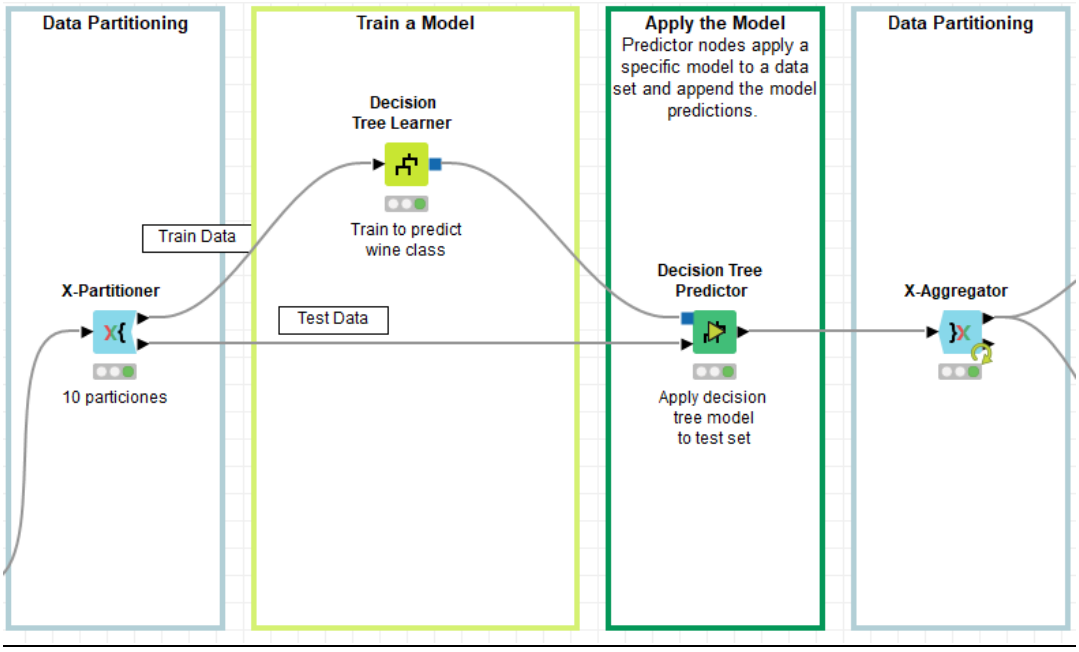
Antes de comenzar la partición de los datos, realizamos el filtro de correlación que hemos visto en el apartado anterior, y tras esto, normalizamos los datos para entrenar con valores entre 0 y 1.

Vamos entonces a ver que resultados obtenemos con cada uno de los modelos y cuál nos ofrece la mayor precisión general y mayor F-Medio en las distintas clases.

DECISION TREE

En la siguiente imagen podemos ver la estructura general usada para todos los modelos entrenados.

1. Particionamos los datos con X-Partitioner (Número de validaciones: 10) con muestreo estratificado.
2. Entrenamos el modelo pasándole los datos de entrenamiento.
3. Pasamos el modelo entrenado al predictor, que recibe también los datos de test.
4. Extremos la salida al X-Aggregator.



8. Estructura de entrnamiento del Árbol de Decisión

En este caso, estamos entrenando un árbol de decisión, que ya adelantamos, es el modelo que menos acierto tiene, con diferencia.

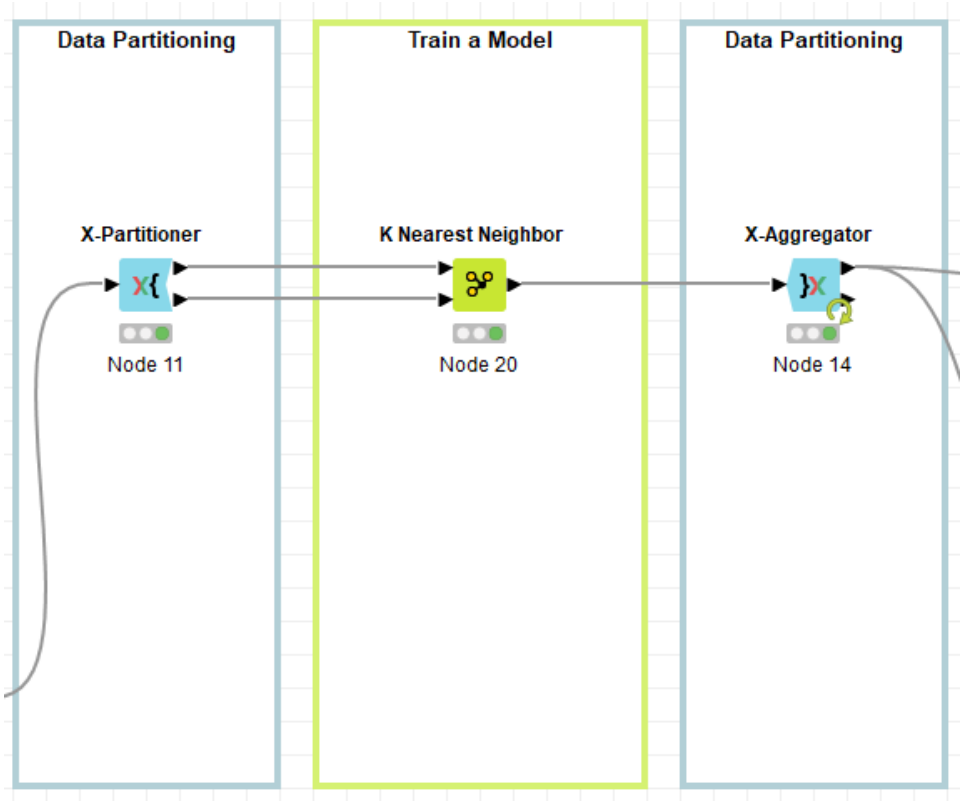
Row ID	tinto	claro	blanco
tinto	54	4	1
claro	1	62	8
blanco	3	7	38

9. Matriz de confusión del Árbol de Decisión

Row ID	TruePo...	FalsePo...	TrueNe...	FalseN...	Recall	Precision	Sensitivity	Specificity	F-meas...	Accuracy	Cohen'...
tinto	54	4	115	5	0.915	0.931	0.915	0.966	0.923	?	?
claro	62	11	96	9	0.873	0.849	0.873	0.897	0.861	?	?
blanco	38	9	121	10	0.792	0.809	0.792	0.931	0.8	?	?
Overall	?	?	?	?	?	?	?	?	?	0.865	0.795

10. Tabla de estadísticas de accuracy del Árbol de Decisión. Nos incluye la matriz de confusión, los valores de F para los tres clases, así como el accuracy y el Cohen's Kappa

K NEAREST (N VECINOS = 3)



11. Estructura del modelo K vecinos. Podemos observar que, a diferencia del anterior modelo, este recibe tanto el dataset de entrenamiento como el dataset de test.

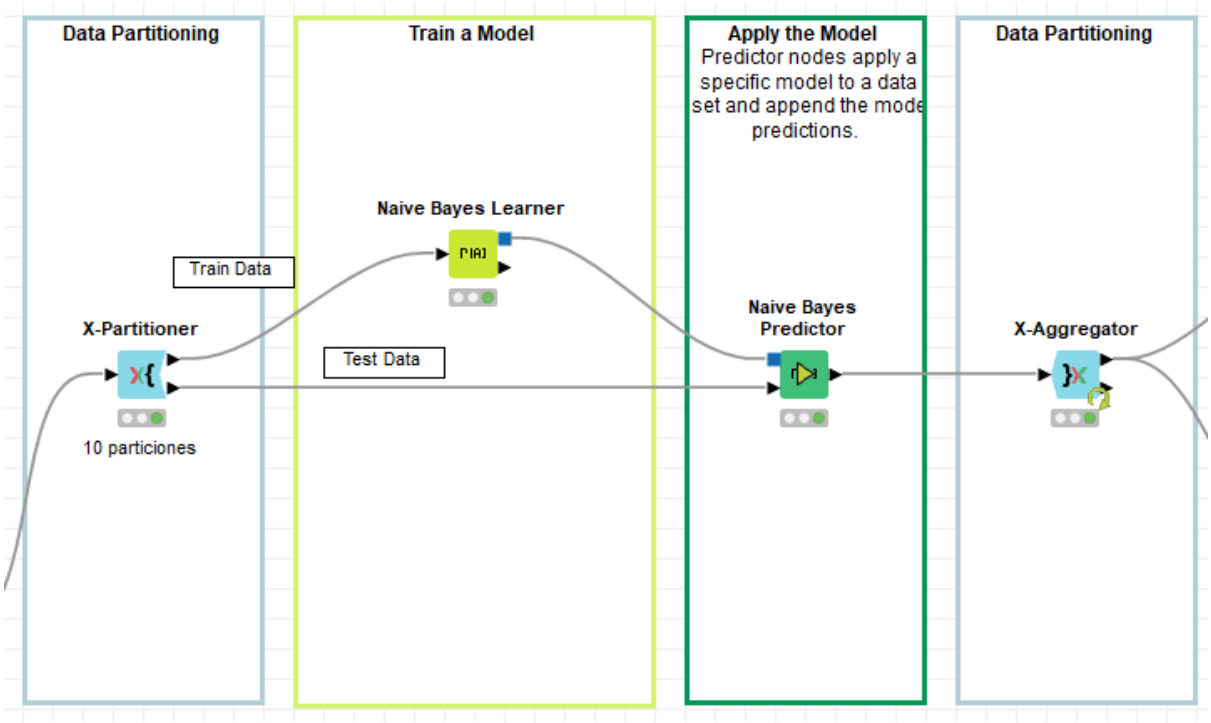
Row ID	tinto	claro	blanco
tinto	59	0	0
claro	1	67	3
blanco	0	0	48

12. Matriz de confusión del modelo K Vecinos

Row ID	TruePo...	FalsePo...	TrueNe...	FalseN...	Recall	Precision	Sensitivity	Specificity	F-meas...	Accuracy	Cohen'...
tinto	59	1	118	0	1	0.983	1	0.992	0.992	?	?
claro	67	0	107	4	0.944	1	0.944	1	0.971	?	?
blanco	48	3	127	0	1	0.941	1	0.977	0.97	?	?
Overall	?	?	?	?	?	?	?	?	?	0.978	0.966

13. Tabla de estadísticas de accuracy.

BAYES



14. Estructura del Clasificador de Bayes.

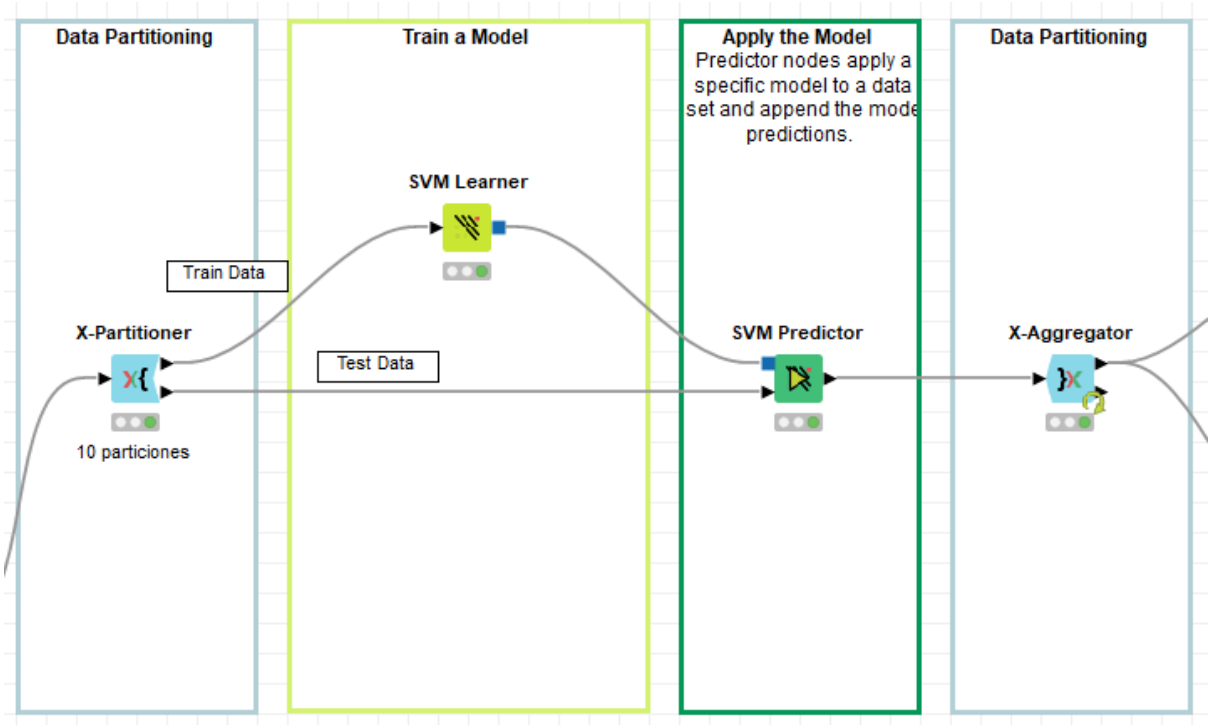
Row ID	tinto	claro	blanco
tinto	57	2	0
claro	2	67	2
blanco	0	0	48

15. Matriz de confusión del modelo.

Row ID	TruePo...	FalsePo...	TrueNe...	FalseN...	Recall	Precision	Sensitivity	Specificity	F-meas...	Accuracy	Cohen'...
tinto	57	2	117	2	0.966	0.966	0.966	0.983	0.966	?	?
claro	67	2	105	4	0.944	0.971	0.944	0.981	0.957	?	?
blanco	48	2	128	0	1	0.96	1	0.985	0.98	?	?
Overall	?	?	?	?	?	?	?	?	?	0.966	0.949

16. Tabla de estadísticas de accuracy.

SVM (SUPPORT-VECTOR MACHINE)



17. Estructura del modelo SVM.

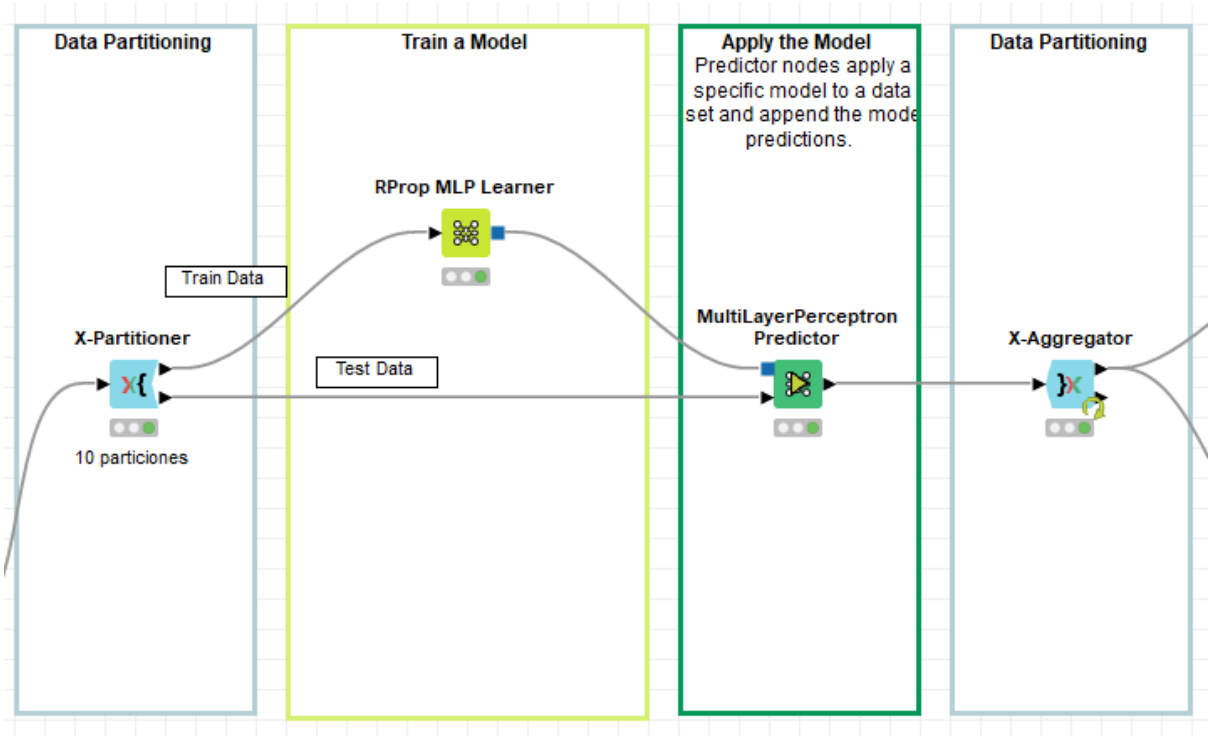
Row ID	tinto	claro	blanco
tinto	58	1	0
claro	2	65	4
blanco	0	0	48

18. Matriz de confusión del modelo.

Row ID	TruePo...	FalsePo...	TrueNe...	FalseN...	Recall	Precision	Sensitivity	Specificity	F-meas...	Accuracy	Cohen'...
tinto	58	2	117	1	0.983	0.967	0.983	0.983	0.975	?	?
claro	65	1	106	6	0.915	0.985	0.915	0.991	0.949	?	?
blanco	48	4	126	0	1	0.923	1	0.969	0.96	?	?
Overall	?	?	?	?	?	?	?	?	?	0.961	0.941

19. Tabla de estadísticas de accuracy.

MLP (MULTILAYER PERCEPTRON)



20. Estructura del modelo MLP.

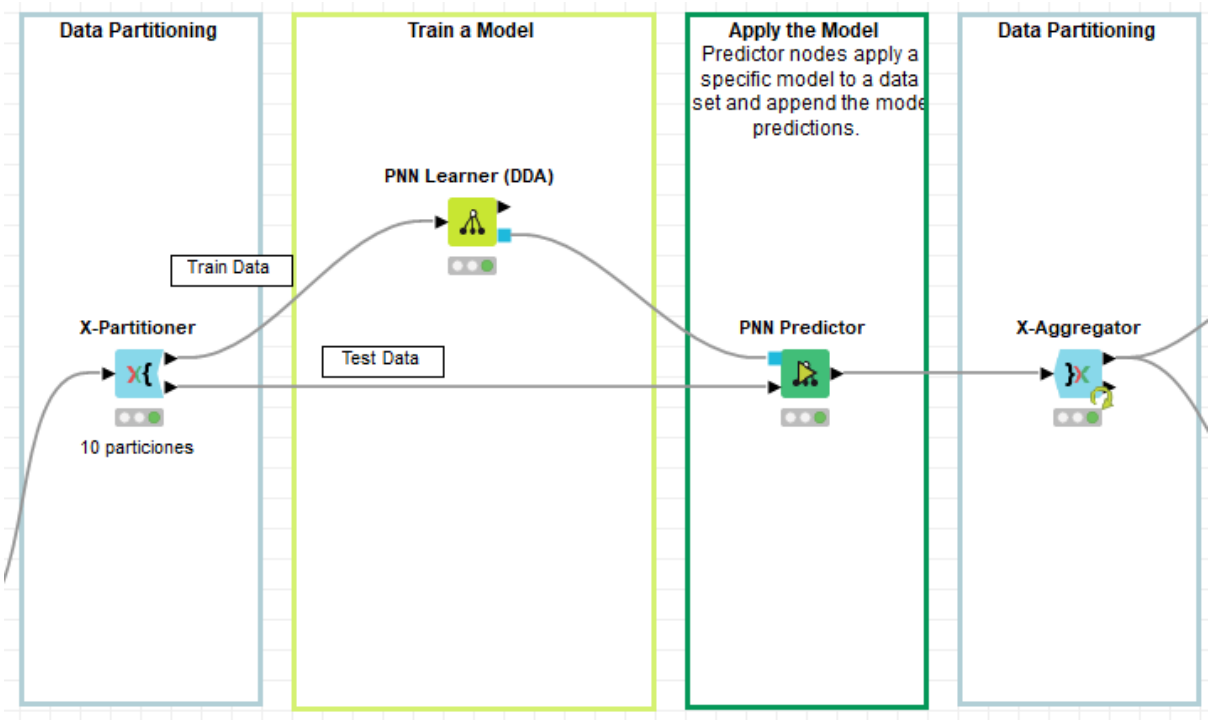
Row ID	tinto	claro	blanco
tinto	57	2	0
claro	1	69	1
blanco	0	1	47

21. Matriz de confusión del modelo.

Row ID	TruePo...	FalsePo...	TrueNe...	FalseN...	Recall	Precision	Sensitivity	Specificity	F-meas...	Accuracy	Cohen'...
tinto	57	1	118	2	0.966	0.983	0.966	0.992	0.974	?	?
claro	69	3	104	2	0.972	0.958	0.972	0.972	0.965	?	?
blanco	47	1	129	1	0.979	0.979	0.979	0.992	0.979	?	?
Overall	?	?	?	?	?	?	?	?	?	0.972	0.957

22. Tabla de estadísticas de accuracy.

PNN (PROBABILISTIC NEURAL NETWORK)



23. Estructura del modelo PNN.

Row ID	tinto	claro	blanco
tinto	59	0	0
claro	9	58	4
blanco	0	0	48

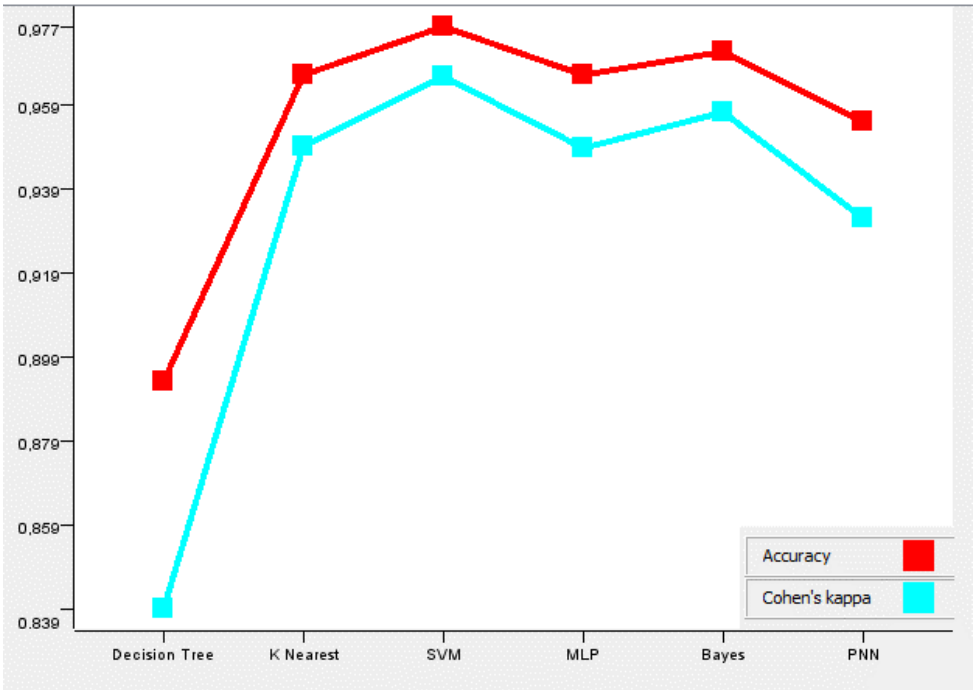
24. Matriz de confusión del modelo.

Row ID	TruePo...	FalsePo...	TrueNe...	FalseN...	Recall	Precision	Sensitivity	Specificity	F-meas...	Accuracy	Cohen'...
tinto	59	9	110	0	1	0.868	1	0.924	0.929	?	?
claro	58	0	107	13	0.817	1	0.817	1	0.899	?	?
blanco	48	4	126	0	1	0.923	1	0.969	0.96	?	?
Overall	?	?	?	?	?	?	?	?	?	0.927	0.89

25. Tabla de estadísticas de accuracy.

CONCLUSIÓN

Tras la ejecución de todos los modelos, tenemos un nodo de convergencia que nos concatena todos los accuracy de los modelos para poder ser mostrados en un gráfico linear.



26. Resumen de accuracy entre los distintos modelos. Vemos como SVM es el modelo que mayor accuracy produce, siendo el Árbol de Decisión el menor.

En la imagen podemos ver los modelos en el eje X. El modelo que ha alcanzado mejores puntuaciones tanto en *accuracy* como en el coeficiente *kappa de Cohen*.

Con este último seremos capaces de hacer una comparación ligeramente más concisa, ya que tiene mejor en cuenta los datos de la matriz de confusión.

Si solamente tenemos en cuenta que estos dos valores, el claro perdedor es el árbol de decisión, pero por otro lado es el más rápido de todos. El claro ganador es el SVM, con 0,972 de accuracy, seguido del K Nearest.