Práctica de Minería de Datos

Curso de Posgrado (Maestría/Doctorado)

Adaptado de la Universitat Politècnica de València

Índice

Esta práctica permite afianzar paso a paso lo visto hasta el momento en el curso utilizando ejemplos muy sencillos y sin realizar validaciones apropiadas de los modelos extraídos. El objetivo de este taller es manejar los operadores básicos de RapidMiner para las tareas de clasificación, clustering y reglas de asociación.

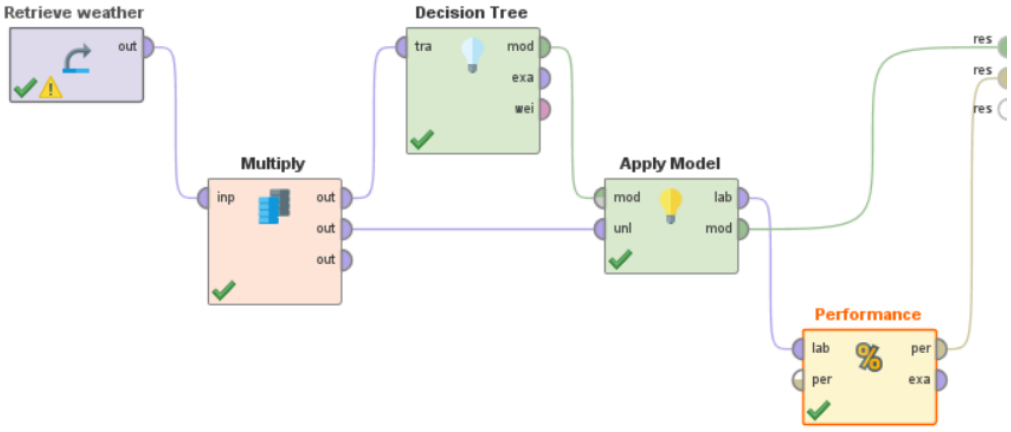
1. Clasificación basada en el data set de weather

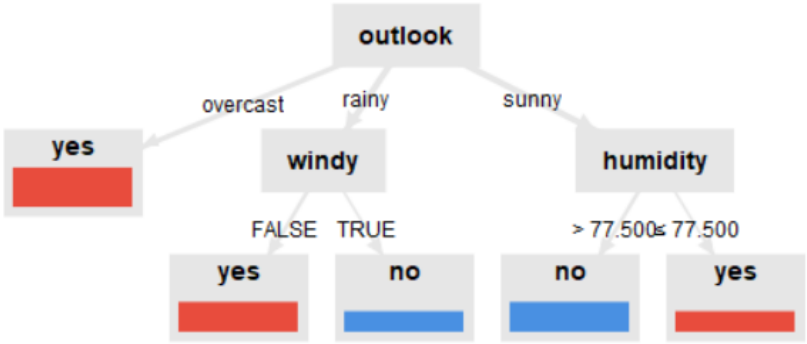
Del curso de "Minería de Datos (2021-01) Maestría - Doctorado" disponible en Moodle (Universidad del Cauca) descargue el data set weather.csv. Este archivo tiene datos acerca de las características (aspectos meteorológicos como vista (outlook), temperatura (temperature), humedad (humidity) y presencia o no de viento (windy) de los días que se ha podido jugar o no a tenis (play). El objetivo de este primer punto es predecir si basado en los datos de un día se podrá o no jugar al tenis. Los datos de este data set son los siguientes:

outlook	temperature	humidity	windy	play
sunny	85.0	85.0	FALSE	no
sunny	80.0	90.0	TRUE	no
overcast	83.0	86.0	FALSE	yes
rainy	70.0	96.0	FALSE	yes
rainy	68.0	80.0	FALSE	yes
rainy	65.0	70.0	TRUE	no
overcast	64.0	65.0	TRUE	yes
sunny	72.0	95.0	FALSE	no
sunny	69.0	70.0	FALSE	yes
rainy	75.0	80.0	FALSE	yes
sunny	75.0	70.0	TRUE	yes
overcast	72.0	90.0	TRUE	yes
overcast	81.0	75.0	FALSE	yes
rainy	71.0	91.0	TRUE	no

Pasos:

1.1. Importar el data set en el repositorio local de RapidMiner.
1.2. En un proceso en blanco, recupere el data set y conéctelo a la salida.
1.3. Realice un análisis estadístico (tipos de datos, valores categóricos, rangos, distribución, datos faltantes, mínimos, máximos, medias, modas, entre otros) y visual de cada atributo y de relaciones entre estos atributos y con la variable objetivo. Use Jitter para separar datos colapsados. Tomar nota de hallazgos interesantes si los hay.
1.4. Incluya los operadores de Árbol de Decisión y Aplicar Modelo y Rendimiento (Clasificación) y conéctelos apropiadamente para obtener el modelo y el rendimiento de dicho modelo y observe los resultados obtenidos (Evaluación usando el mismo data set de entrenamiento).

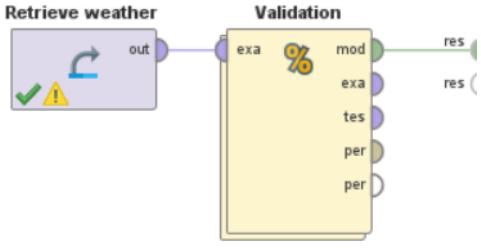




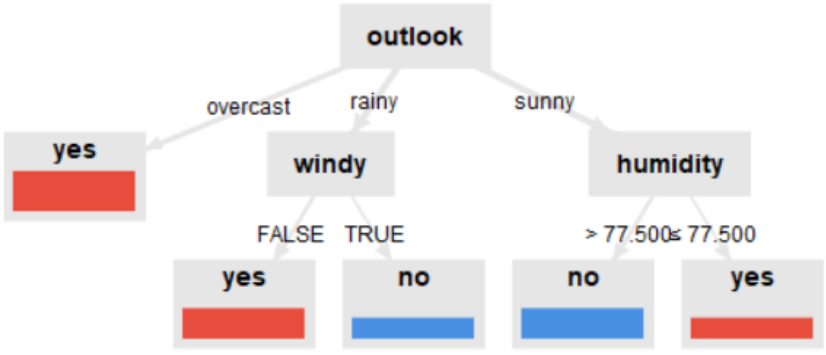
accuracy: 100.00%

	true no	true yes	class precision
pred. no	5	0	100.00%
pred. yes	0	9	100.00%
class recall	100.00%	100.00%	

1.5. Incluya un operador de bloque de validación cruzada nominal (por defecto usa un Árbol de Decisión como clasificador, y los operadores de Aplicar modelo y Rendimiento de clasificación) conéctelo apropiadamente para obtener el modelo, el rendimiento de dicho modelo y observe los resultados obtenidos. ¿Entiende porque el rendimiento baja?, ¿cuántos modelos se generan para obtener el valor de rendimiento con validación cruzada? y ¿cuál de los modelos generados es el que presenta al final?



A cross-validation evaluating a decision tree model.



accuracy: 35.00% +/- 41.16% (micro average: 42.86%)

	true no	true yes	class precision
pred. no	2	5	28.57%
pred. yes	3	4	57.14%
class recall	40.00%	44.44%	

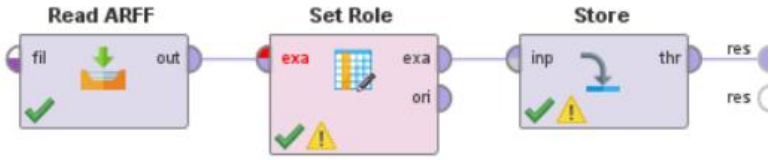
2. Clasificación basada en el data set de cars

En este caso se trata de predecir donde fue construido (Brand como campo nominal con tres valores Japón, Europa o Estados Unidos) un automóvil basado en algunos datos de este como son: millas por galón de gasolina (mpg: real o continuo), número de cilindros en el motor (cylinders: nominal), pulgadas cubicas del motor (cubicinches: real), caballos de fuerza (hp: real), peso en libras (weightlbs: real), tiempo para llegar a 60 millas por minuto (time-to-60: nominal) y año de construcción (year: real). A continuación, una muestra de los 261 registros o instancias que componen el data set.

Row No.	mpg	cylinders	cubicinches	hp	weightlbs	time-to-60	year	brand
1	14	8	350	165	4209	12	1972	US
2	31.900	4	89	71	1925	14	1980	Europe
3	17	8	302	140	3449	11	1971	US
4	15	8	400	150	3761	10	1971	US
5	30.500	4	98	63	2051	17	1978	US
6	23	8	350	125	3900	17	1980	US
7	13	8	351	158	4363	13	1974	US
8	14	8	440	215	4312	9	1971	US
9	25.400	5	183	77	3530	20	1980	Europe
10	37.700	4	89	62	2050	17	1982	Japan
11	34	4	108	70	2245	17	1983	Japan
12	34.300	4	97	78	2188	16	1981	Europe
13	16	8	302	140	4141	14	1975	US
14	11	8	350	180	3664	11	1974	US

Pasos:

2.1. Usando los operadores de Leer ARFF (es posible que requiera incluir la extensión de Weka), de Fijar Role y de Almacenar, lea, defina como etiqueta la columna brand y almacene el data set cars.arff en su repositorio local de RapidMiner.



2.2. En un proceso en blanco, recupere el data set y conéctelo a la salida.

2.3. Realice un análisis estadístico (tipos de datos, valores categóricos, rangos, distribución, datos faltantes, mínimos, máximos, medias, modas, entre otros) y visual de cada atributo y de relaciones entre estos atributos y con la variable objetivo. Use Jitter para separar datos colapsados. Tomar nota de hallazgos interesantes si los hay.

2.4. Usando un operador de bloque de validación cruzada nominal y el clasificador por Defecto tome una decisión. Recuerde que debe usar la moda y no la media ya que es un problema de clasificación. ¿Cómo se toma esta decisión?

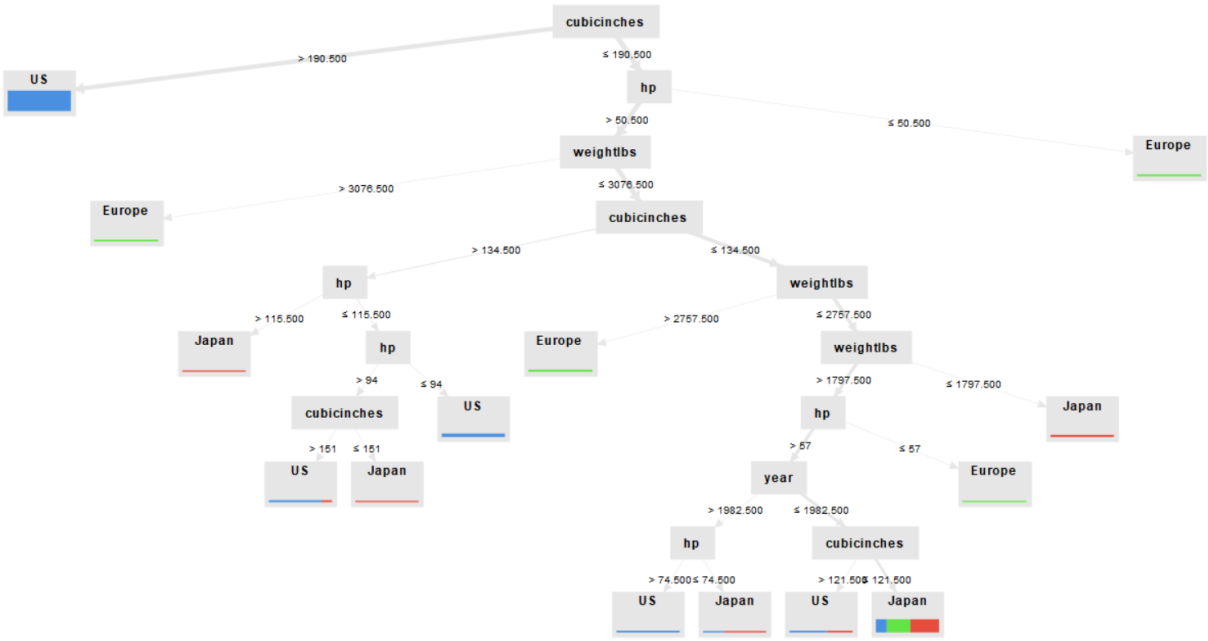
Default

```
Default (prediction model for label brand)
default value: US
```

accuracy: 62.07% +/- 1.25% (micro average: 62.07%)

	true US	true Europe	true Japan	class precision
pred. US	162	48	51	62.07%
pred. Europe	0	0	0	0.00%
pred. Japan	0	0	0	0.00%
class recall	100.00%	0.00%	0.00%	

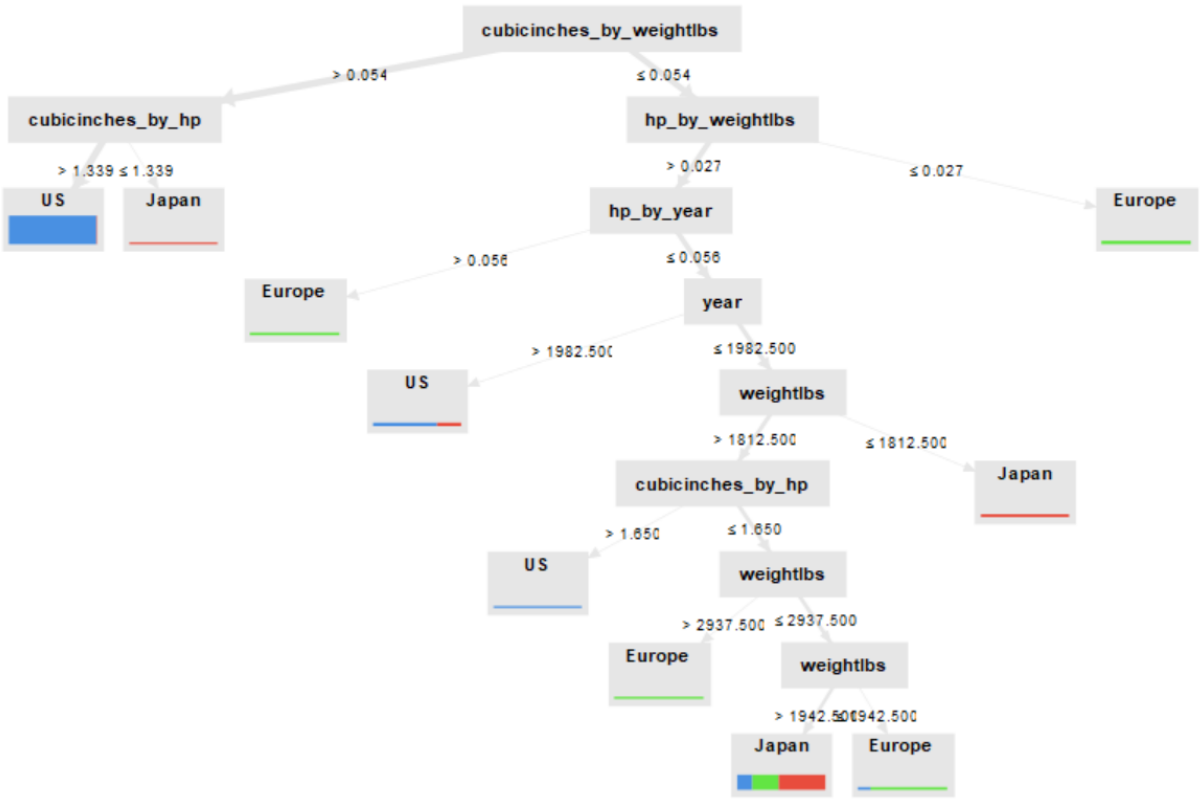
2.5. Usando un operador de bloque de validación cruzada nominal y el clasificador de Árbol de Decisión. ¿Qué modelo obtiene?, ¿Cuál es el rendimiento del modelo?, ¿Este modelo es mejor que el modelo por defecto? y ¿Por qué?



accuracy: 74.69% +/- 11.40% (micro average: 74.71%)

	true US	true Europe	true Japan	class precision
pred. US	143	6	10	89.94%
pred. Europe	4	21	10	60.00%
pred. Japan	15	21	31	46.27%
class recall	88.27%	43.75%	60.78%	

2.6. Genere cuatro nuevos atributos (cubicinches_by_weightlbs, cubicinches_by_hp, hp_by_weightlbs, hp_by_year) usando un operador de bloque de validación cruzada nominal y el clasificador de Árbol de Decisión. ¿Qué modelo obtiene?, ¿Cuál es el rendimiento del modelo?, ¿Este modelo es mejor que el modelo anterior?, ¿Tiene sentido? ¿Cómo se imagina que se obtuvieron esas combinaciones de atributos? y ¿si los tuviera que generar automáticamente como lo haría? (pista operador Loop Attributes).



accuracy: 77.78% +/- 8.65% (micro average: 77.78%)

	true US	true Europe	true Japan	class precision
pred. US	145	3	7	93.55%
pred. Europe	1	19	5	76.00%
pred. Japan	16	26	39	48.15%
class recall	89.51%	39.58%	76.47%	

3. Filtrado de atributos

En el árbol de decisión final se puede observar que no se utilizan todos los atributos para efectuar una clasificación, esto indica que hay atributos que no son significativos para la resolución del problema. Existen métodos como los árboles de decisión, a los cuales no les afecta de manera grave la presencia de atributos no significativos, ya que en el propio mecanismo de aprendizaje realizan una selección de atributos por su relevancia. Sin embargo, otros métodos no realizan este proceso, por lo que si se realiza un filtrado de atributos previo al aprendizaje se puede mejorar de manera relevante su precisión, y al mismo tiempo la simplicidad del modelo obtenido.

Pasos:

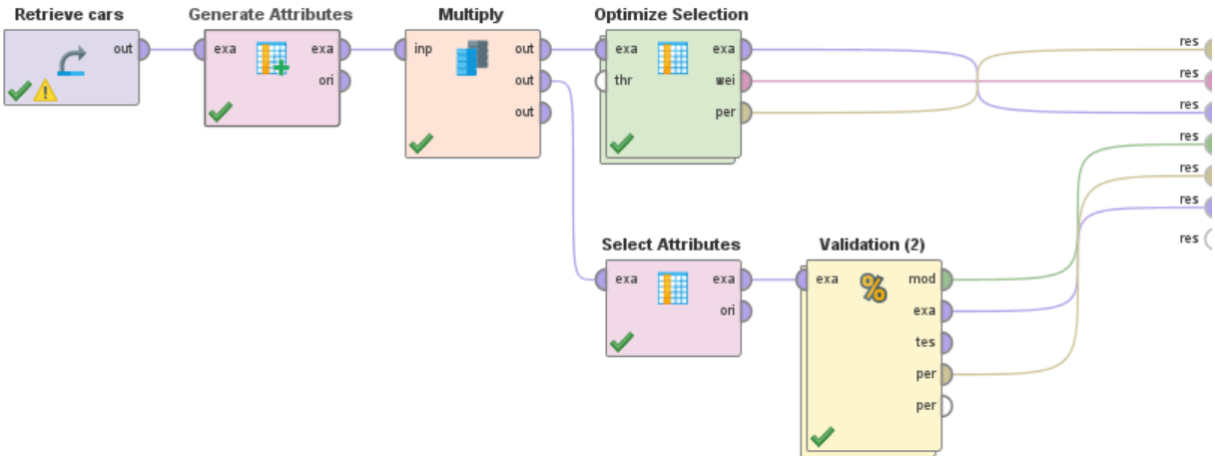
3.1. Basado en el proceso definido en el punto anterior (punto 2.6) Cambie el clasificador por Naive Bayes (un ejemplo de método de aprendizaje que reduce su *calidad* ante la presencia de atributos no relevantes), ejecute el proceso y revise los resultados. ¿Ese clasificador hace selección de atributos como la hizo el Árbol de Decisión? ¿Cuál de los dos modelos es más simple y presenta mejor rendimiento (medido en exactitud o accuracy)?

accuracy: 71.65% +/- 7.71% (micro average: 71.65%)

	true US	true Europe	true Japan	class precision
pred. US	124	5	4	93.23%
pred. Europe	7	21	5	63.64%
pred. Japan	31	22	42	44.21%
class recall	76.54%	43.75%	82.35%	

Attribute	Parameter	US	Europe	Japan
mpg	mean	19.625	27.508	30.218
mpg	standard deviation	6.240	6.994	6.123
cylinders	value=8	0.469	0.000	0.000
cylinders	value=4	0.235	0.895	0.862
cylinders	value=5	0.000	0.063	0.000
cylinders	value=6	0.296	0.042	0.098
cylinders	value=3	0.000	0.000	0.039
cylinders	value=unknown	0.000	0.000	0.000
cubicinches	mean	259.019	108.583	104.216
cubicinches	standard deviation	99.533	22.419	24.747
hp	mean	122.698	79.958	79.314

3.2. Basado en el operador que Optimiza la Selección (Selección de atributos) y usando el clasificador Naive Bayes con validación cruzada (defina el uso de una semilla aleatoria local - local random seed - en un valor, por ejemplo, el valor por defecto de 1992), ejecute el proceso mostrando como salida el data set resultante, los pesos de los atributos y el rendimiento del clasificador. Luego, agregue un operador de selección de atributos manual y escoja los atributos seleccionados automáticamente, agregue una validación cruzada (igual semilla, 1992) con Naive Bayes y revise si obtiene los mismos resultados. ¿Qué sucede si no usa las semillas aleatorias de la validación cruzada?, ¿los resultados son iguales? y ¿por qué?



Attribute	Parameter	US	Europe	Japan
time-to-60	value=12	0.086	0.021	0.000
time-to-60	value=14	0.093	0.146	0.098
time-to-60	value=11	0.062	0.000	0.020
time-to-60	value=10	0.037	0.000	0.000
time-to-60	value=17	0.099	0.083	0.235
time-to-60	value=13	0.148	0.021	0.020
time-to-60	value=9	0.019	0.000	0.000

accuracy: 77.41% +/- 7.30% (micro average: 77.39%)

	true US	true Europe	true Japan	class precision
pred. US	138	4	9	91.39%
pred. Europe	9	34	12	61.82%
pred. Japan	15	10	30	54.55%
class recall	85.19%	70.83%	58.82%	

Row No.	brand	time-to-60	cubicinches_by_weightlbs ↑
38	Japan	14	0.029
157	Japan	14	0.033
173	Europe	20	0.036
214	Europe	20	0.038
107	Europe	24	0.039

4. Aprendizaje sensible al costo

En muchos problemas, nos encontraremos situaciones donde el tipo de error que se comente define un costo diferente. Por ejemplo, en un entorno bancario se cuenta con un modelo que recomienda si conceder o no un crédito a un determinado cliente a partir de las características propias de ese cliente. Obviamente, y desde el punto de vista del banco, es mucho más costoso que el sistema se equivoque dando un crédito a una persona que no lo devuelve, que la situación contraria, denegar un crédito a un cliente que sí lo devolvería.

Para este tipo de problemas, la información sobre los costos de los errores viene expresada a través de una matriz de costos. En este tipo de matrices se recoge el costo de cada una de las posibles combinaciones entre la case predicha por el modelo y la clase real.

RapidMiner ofrece mecanismos para evaluar modelos con respecto al costo de clasificación, así como de métodos de aprendizaje basados en reducir el costo en vez de incrementar la cantidad de instancias correctamente clasificadas.

En este ejemplo se utilizará el data set “credit-g.csv”. Este conjunto de datos contiene 1.000 ejemplos que representan clientes de una entidad bancaria que solicitaron y recibieron un crédito. Existen siete atributos numéricos y 13 nominales. Los atributos de cada registro indican información sobre el cliente en cuestión así como información del crédito, a saber: estado de la cuenta corriente (checking_status), duración en meses de la cuenta (duration), historial crediticio (credit_history con valores relacionados con créditos tomados, pagados debidamente, retrasos, cuentas críticas), propósito del crédito (purpose), cantidad solicitada del crédito (credit_amount), estado de la cuenta de ahorro o de bonos (savings_status), empleo actual, en número de años (employment), cuota en porcentaje del ingreso disponible (installment_commitment), estado personal como casado, soltero, entre otros (personal_status), otros deudores/garantías (other_parties), residencia actual desde X años (residence_since), propiedades, por ejemplo, bienes raíces, autos, entre otros (property_magnitude), edad en años (age), otros deudas a bancos, tiendas u otros (other_payment_plans), vivienda en alquiler, propia, u otra (housing), número de créditos existentes en este banco (existing_credits), tipo de trabajo (job), número de personas dependientes (num_dependents), posee teléfono o no (own_telephone), es trabajador extranjero o n) (foreign_worker), y si pago bien su crédito (good) o no (bad) que corresponde a la clase (class).

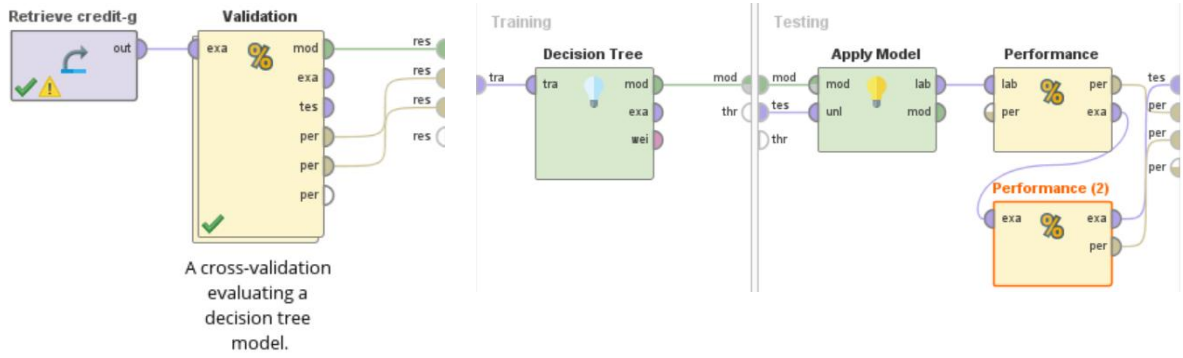
Este data set, trae información de los costes de clasificación errónea, en concreto la siguiente matriz de costos:

		Real
	True good	True bad
Predicted good	0	5
Predicted bad	1	0

Esta tabla indica que es 5 veces más costoso si se otorga un crédito a una persona que no lo devuelve, que la situación contraria.

Pasos:

4.1. Importar el data set en el repositorio local de RapidMiner.
4.2. En un proceso en blanco, recupere el data set y conéctelo a la salida.
4.3. Realice un análisis estadístico (tipos de datos, valores categóricos, rangos, distribución, datos faltantes, mínimos, máximos, medias, modas, entre otros) y visual de cada atributo y de relaciones entre estos atributos y con la variable objetivo. Use Jitter para separar datos colapsados. Tomar nota de hallazgos interesantes si los hay.
4.4. Incluya un operador de bloque de validación cruzada nominal (con un Árbol de Decisión y los operadores de Aplicar modelo y Rendimiento de clasificación) conéctelo apropiadamente para obtener el modelo, el rendimiento de dicho modelo y observe los resultados obtenidos. Luego, incluya además dentro de la validación cruzada un operador Rendimiento basado en costos con los datos de la matriz de costos presentada previamente y muestre ese resultado al final del proceso. ¿Cuál es el costo de las decisiones que toma el clasificador?, ¿cuál es el rendimiento en accuracy? y ¿el clasificador está tomando las decisiones por costo o por cantidad de instancias correctamente clasificadas?



Misclassificationcosts

```
Misclassificationcosts: 1.137 +/- 0.273 (micro average: 1.137)
```

accuracy: 68.70% +/- 3.89% (micro average: 68.70%)

	true good	true bad	class precision
pred. good	593	206	74.22%
pred. bad	107	94	46.77%
class recall	84.71%	31.33%	

4.5. En el operador de bloque de validación cruzada nominal cambie el Árbol de Decisión por un operador MetaCost, configúrele la matriz de costos y dentro agregue un Árbol de decisiones y deje el resto del proceso igual. ¿Cuál es el costo de las decisiones que toma el clasificador?, ¿cuál es el rendimiento en accuracy? ¿el clasificador está tomando las decisiones por costo o por cantidad de instancias correctamente clasificadas? Y ¿Entre el modelo del punto anterior y el obtenido en este punto cual seleccionaría y por qué?

accuracy: 48.10% +/- 8.46% (micro average: 48.10%)

	true good	true bad	class precision
pred. good	211	30	87.55%
pred. bad	489	270	35.57%
class recall	30.14%	90.00%	

Misclassificationcosts

```
Misclassificationcosts: 0.639 +/- 0.048 (micro average: 0.639)
```

4.6. En el operador MetaCost cambie el Árbol de decisiones por el operador Naive Bayes. ¿Cuál es el costo de las decisiones que toma el clasificador?, ¿cuál es el rendimiento en accuracy? Y ¿Entre el modelo del punto anterior y el obtenido en este punto cual seleccionaría y por qué?

accuracy: 68.90% +/- 6.10% (micro average: 68.90%)

	true good	true bad	class precision
pred. good	445	56	88.82%
pred. bad	255	244	48.90%
class recall	63.57%	81.33%	

Misclassificationcosts

```
Misclassificationcosts: 0.535 +/- 0.138 (micro average: 0.535)
```

5. Combinación de modelos

Uno de los aspectos más destacados de RapidMiner es la gran cantidad de métodos de combinación de modelos que posee. Estos métodos mejoran la calidad de las predicciones mediante la combinación de las predicciones de varios modelos. Par lograr esto, los métodos implementan la estrategia de "cuatro ojos ven más que dos", aunque también es cierto que para que esta afirmación se cumpla los ojos deben tener buena vista, y además no deben tener un comportamiento idéntico, ya que en ese caso no habría mejora.

Por el contrario, utilizar combinar modelos también tiene desventajas, principalmente, el modelo se hace más complejo, perdiendo interpretabilidad y explicabilidad. Además, se necesitan más recursos (tiempo y memoria) para el aprendizaje de los múltiples modelos que se esperan combinar. En este ejemplo se seguirá usando el data set "credit-g.csv".

Pasos:

5.1. En un proceso en blanco, recupere el data set y conéctelo a la salida.
5.2. Incluya un operador de bloque de validación cruzada nominal (con un Árbol de Decisión y los operadores de Aplicar modelo y Rendimiento de clasificación) conéctelo apropiadamente para obtener el modelo, el rendimiento de dicho modelo y observe los resultados obtenidos. En este punto cuenta con un modelo que reporta un accuracy de 68.70%. Cambie los siguientes parámetros de Árbol de decisión, criterion = gini_index, maximal_depth = 12 y minimal_leaf_size = 20. Con estos parámetros obtendrá un accuracy de 73.60%. Estos parámetros se obtuvieron con una optimización sencilla por grilla con el operador Optimize Parameters (Grid).
5.3. En el operador de bloque de validación cruzada nominal cambie el Árbol de Decisión por un operador Bagging (Esta técnica se fundamenta en construir un conjunto de n modelos mediante el aprendizaje desde n conjuntos de datos. Cada conjunto de datos se construye realizando un muestreo con repetición del conjunto de datos de entrenamiento.) con 40 iterations (modelos) y dentro del bagging deje el Árbol de decisión con la siguiente configuración de parámetros, criterion = accuracy, maximal_depth = 12 y minimal_leaf_size = 20. ¿Cuál es el rendimiento en accuracy? y ¿Entre el modelo del punto anterior y el obtenido en este punto cual seleccionaría y por qué?

accuracy: 72.50% +/- 3.66% (micro average: 72.50%)

	true good	true bad	class precision
pred. good	615	190	76.40%
pred. bad	85	110	56.41%
class recall	87.86%	36.67%	

5.4. En el operador de bloque de validación cruzada nominal cambie el operador Bagging por AdaBost (Esta técnica es bastante parecida al Bagging, aunque utiliza una estrategia más ingeniosa ya que cada iteración intenta corregir los errores cometidos anteriormente dando más peso a los datos que se han clasificado erróneamente. Esta forma de trabajar dificulta que su implementación se ejecute en paralelo) con 40 iterations (modelos) y dentro del boosting deje el Árbol de decisión con la siguiente configuración de parámetros, criterion = gini_index, maximal_depth = 6 y minimal_leaf_size = 20. ¿Cuál es el rendimiento en accuracy? ¿Entre el modelo del punto anterior y el obtenido en este punto cual seleccionaría y por qué? E investigue a fondo cómo funciona un ensamblado por boosting tanto en el entrenamiento como en la predicción.

accuracy: 74.30% +/- 2.87% (micro average: 74.30%)

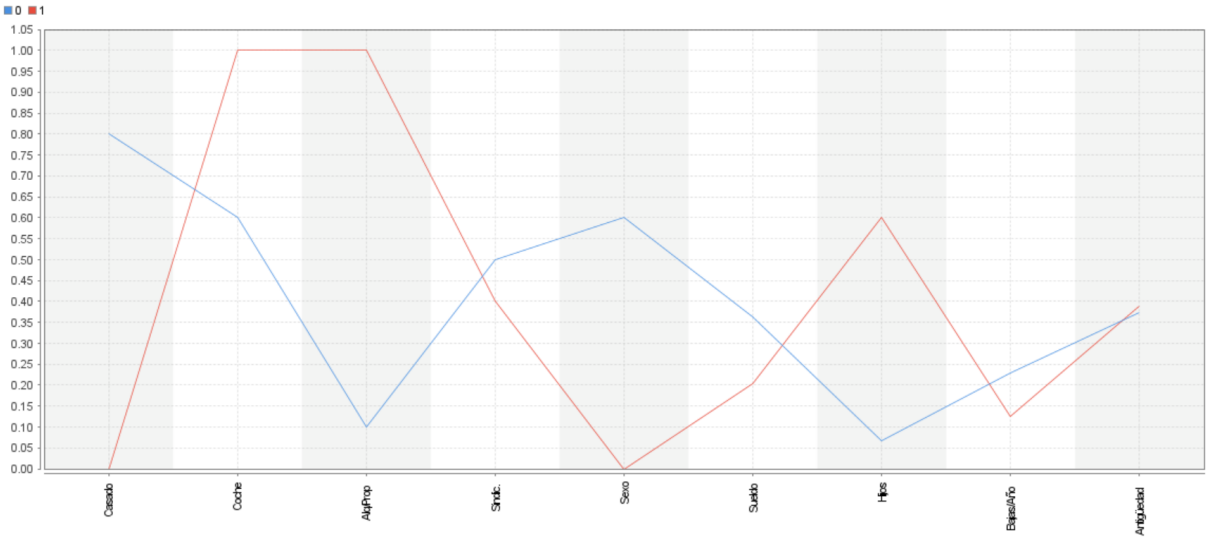
	true good	true bad	class precision
pred. good	619	176	77.86%
pred. bad	81	124	60.49%
class recall	88.43%	41.33%	

6. Agrupamiento o clustering

La empresa de software para Internet “Memolum Web” quiere extraer tipologías de empleados, con el objetivo de hacer una política de personal más fundamentada y seleccionar a qué grupos incentivar.
Las variables que se recogen de las fichas de los 15 empleados de la empresa son: Sueldo: sueldo anual en euros, Casado: si está casado o no, Coche: si viene en coche a trabajar (o al menos si lo aparca en el párking de la empresa), Hijos: el número de hijos, Alq/Prop: si vive en una casa alquilada o propia, Sindic.: si pertenece al sindicato de la Internet, Bajas/Año: media del número de bajas por año, Antigüedad: antigüedad en la empresa (en años), y Sexo: H: hombre, M: mujer. Los datos de los 15 empleados se encuentran en el fichero “empleados.csv”. Se intenta extraer grupos de entre estos quince empleados.

Pasos:

6.1. Importar el data set en el repositorio local de RapidMiner. Asigne los tipos de datos entero y binomial a los campos según corresponda.
6.2. En un proceso en blanco, recupere el data set y conéctelo a la salida.
6.3. Realice un análisis estadístico (tipos de datos, valores categóricos, rangos, distribución, datos faltantes, mínimos, máximos, medias, modas, entre otros) y visual de cada atributo y de relaciones entre estos atributos y con la variable objetivo. Use Jitter para separar datos colapsados. Tomar nota de hallazgos interesantes si los hay.
6.4. Para tratar con los datos en el mismo rango se realizarán varias operaciones. Primero se mapearan todos los atributos nominales a números de la siguiente forma No (0), Si o Sí (1), H de Hombre (0) y M de Mujer (1), Alquiler (0) y Prop de casa propia (1) usando el operador Map. Segundo, usando el operador Nominal to Numerical con el parámetro coding type = unique integer se modifica el tipo de datos de los campos nominales a numéricos sin crear nuevas columnas como si sucede con la codificación dummy. Tercero, Normalizamos todos los atributos para que queden en el rango de cero a 1 usando el operador Normalize y el parámetro method = range transformation. Finalmente agregamos el operador de Clustering (k-Means) al cual se le asignará el valor de k en 2 para armar dos grupos. Ejecute el proceso y revise el Plor del Modelo de Clustering.



Como se puede apreciar un grupo (rojo) está formado principalmente por Hombres (Sexo 0), No (0) casados que Si (1) tienen coche, SI tienen casa propia (1) y tienen varios hijos (se debe volver a los datos originales para determinar cuántos en promedio). Por el otro lado el grupo (azul) está formado en un 80% por personas Casadas (1) que No (0) tiene casa propia, está repartido entre mujeres y hombres que casi no tienen hijos.

6.5. Realice el mismo análisis con k=3 y comente cual sería una mejor forma de agrupar, con k=2 o con k=3 y ¿por qué?