Telefónica | EDUCACIÓN DIGITAL

Módulo 4:

Soluciones de Machine Learning: Aprendizaje No Supervisado

Módulo 4

Soluciones de Machine Learning: Aprendizaje No Supervisado

El aprendizaje no supervisado es aquél que no requiere de ningún etiquetado previo de las instancias. Se basa en los datos tal y como los recibe y su objetivo es determinar relaciones de similitud, diferencia o asociación.

Aprendizaje no supervisado (*Unsupervised Learning*)

Los modelos que se catalogan bajo este epígrafe son:

- **Clusters** Estos modelos buscan aquellas instancias que son similares entre sí y distintas de las demás para formar agrupaciones de los datos llamadas *clusters*. Dichos modelos nos permitirán predecir a qué cluster corresponde un nuevo elemento. Veremos algún ejemplo de clustering aplicado al sector inmobiliario, para segmentar la oferta de pisos y ver qué productos son similares, y al comercio, creando agrupaciones de tipos de cervezas según los gustos de nuestros clientes.
- **Detectores de anomalías** Estos modelos buscan aquellas instancias que son distintas de la tónica general del resto de datos. El ejemplo que veremos de este tipo de modelos es su aplicación al fraude en el sector de los préstamos bancarios, pero también podrían usarse para limpiar aquellas instancias de un dataset que contienen datos erróneos o muy sesgados (*outliers*).
- **Buscadores de asociaciones** Estos modelos buscan las relaciones existentes entre diversos valores de los campos proporcionados. Las relaciones, o reglas de asociación (*association rules*) son de la forma: cuando este campo tiene este valor entonces, en general, aquel campo tiene aquel valor. El ejemplo típico de este tipo de modelos es el análisis de la cesta de la compra, donde se descubren los productos que se adquieren a la vez más a menudo de lo que sería de esperar si se compraran al azar.

En general, un problema de Machine Learning necesitará aplicar combinaciones de varios tipos de modelos y refinar el proceso hasta llegar a los resultados deseados. Quizás debamos aplicar un detector de anomalías a nuestros datos para limpiar el dataset de aquellas instancias que contengan datos erróneos antes de usarlo para construir un modelo de clasificación. También podemos querer aplicar primero un clustering sobre los datos y crear modelos distintos para cada cluster para que estén más adaptados a los grupos existentes en nuestro datos. Así pues, el proceso hasta llegar a la solución del problema conllevará en general el uso de más de un tipo de modelo.

Veamos algunos ejemplos de problemas que se solucionan con algoritmos de apredizaje no supervisado.

4.1. Segmentación: Clusters

En los problemas de segmentación o *clustering* el objetivo que se persigue es estructurar nuestros datos en grupos según su similitud. Cada grupo (o *cluster*) contendrá un número variable de instancias de nuestro dataset que se consideran similares entre sí porque los valores de los atributos que las describen lo son. A su vez, nos interesará conseguir que los datos agrupados en un *cluster* sean claramente distintos de los que se agrupan en otro *cluster*.

¿Cómo calculamos si nuestros datos son similares?

Para poder decidir cómo se agrupan nuestros datos, los algoritmos usados se basan en la definición de una *distancia* que mide cuán similares son sus atributos. Si pensamos en atributos *numéricos*, la fórmula para

calcular dicha distancia será la usual (distancia euclídea[1]), de forma que cuanto menos difieren los valores de sus atributos, más similares serán las instancias y más probable será que se agrupen.

Otro punto a tener en cuenta para calcular la distancia es cómo contribuyen los campos no numéricos. En el caso de los *categóricos*, se puede convenir que sólo contribuyen cuando las categorías de los dos casos comparados difieren. De forma similar, se debe definir cómo comparar los campos de *texto* o *items*. Para dichos campos, se analiza la frecuencia de sus términos y se comparan usando una técnica llamada similitud coseno[2] (*cosine similarity*). Cuanto más parecido es el conjunto de frecuencias de sus palabras, menos distancia habrá entre las instancias comparadas.

En cualquier caso, debemos tener en cuenta que los rangos en que se mueven los valores de diferentes atributos pueden ser muy distintos. Si estamos comparando viviendas, podemos tener atributos como el número de habitaciones, cuyos valores están por debajo de las decenas y otros como el precio, que estará en los centenares de miles. Para evitar que las diferencias en un atributo predominen sobre los demás, hay que escalar previamente los atributos y asegurar que sus valores estén dentro de un rango común. En la plataforma que usamos, estos cálculos, incluido el escalado de los atributos, ser realizan automáticamente al crear los *clusters*. Nosotros podemos, sin embargo, decidir cambiar ese escalado para conseguir que un atributo concreto tenga más o menos importancia a la hora de establecer la similitud de los datos.

Finalmente, hay que comentar el caso de las instancias con campos numéricos que no tienen un valor asignado. No podemos calcular distancias cuando no tenemos dos valores numéricos a comparar. Por eso, en el caso de que algunas instancias tengan campos vacíos, o se establece un valor por defecto, como cero, el mínimo, la media, etc. o dichas instancias serán descartadas.

Tanto el escalado como los valores numéricos usados por defecto son parámetros que pueden ser configurados en la *pantalla de configuración de cluster*, tal como se muestra en la Figura 4.11.

¿Sabemos el número de grupos a generar?

En ocasiones conocemos el número de clusters que queremos obtener como solución a nuestro problema de clustering. Por ejemplo, si nuestro negocio es la confección de ropa, será útil agrupar las medidas de nuestros clientes en tres clusters que nos permitan elegir las tres tallas que queremos comercializar. Otras veces en cambio, no tendremos ninguna idea a priori sobre el número de grupos a obtener y preferiremos que sea el algoritmo quien lo determine en función de la propia distribución de los datos. En cada caso usaremos un algoritmo distinto.

Número de grupos desconocido: *g-means*

Cuando no tenemos un número de grupos determinado en qué queramos organizar nuestros datos, podemos dejar que el propio algoritmo decida ese número. La decisión se toma en función de si, partiendo un grupo existente, los nuevos subgrupos contienen distribuciones normales de los datos. El algoritmo adecuado en estos casos es *g-means*.

Para entender cómo se hace la agrupación, describiremos a grandes rasgos el funcionamiento de *g-means*. Imaginemos que cada fila de nuestro fichero corresponde a un punto en un espacio donde cada coordenada es uno de sus atributos. Inicialmente partimos de un solo grupo (k=*1*) que contiene todos los puntos (es decir, todas las instancias) y lo dividimos en dos. Después se busca el centro de cada subgrupo de instancias y se unen los dos centros con una recta. Se proyecta cada uno de los puntos sobre esa recta, y se compara la distribución formada por esas proyecciones con el perfil de una Gausiana. Si encajan, consideraremos que es mejor separarlos, y aumentaremos el número de grupos (k). En caso contrario, se mantiene el grupo inicial. El proceso se aplica de nuevo a los nuevos grupos creados hasta que el número de grupos se hace estable. Esos serán los k clusters generados.

[1] https://es.wikipedia.org/wiki/Distancia_euclidiana
[2] https://es.wikipedia.org/wiki/Similitud_coseno

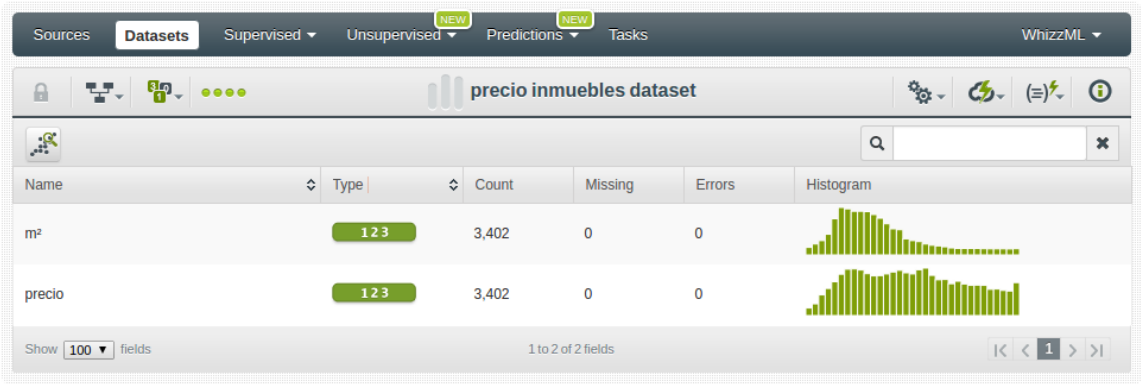


Figura 4.1: Dataset de precios de pisos. Contiene la superfície en metros cuadrados y su precio.

El algoritmo de *g-means* es el usado por defecto cuando creamos un *Cluster* desde un *Dataset*. Como ejemplo sencillo podemos ver cómo se agrupan los datos sobre inmuebles según su superfície y precio. Partimos de un *Dataset* que contene esos datos, como el de la Figura 4.1 y crearemos un *Cluster* con la opción *1-click Cluster* del menú de acciones del *Dataset* tal como vemos en la Figura 4.2.

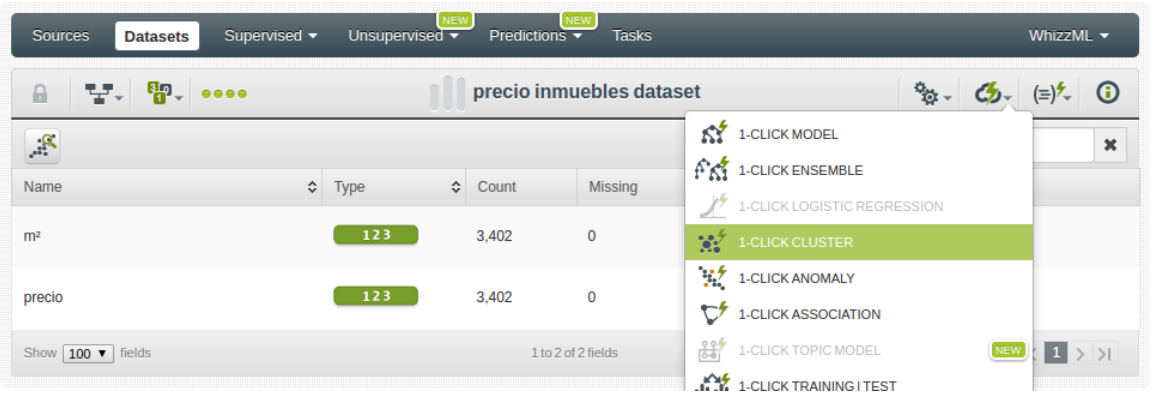


Figura 4.2: Creación de un *Cluster* en un solo clic. El algoritmo usado es *g-means* y todas las opciones son por defecto.

La representación gráfica de los *clusters* nos presenta cada uno de ellos como un círculo coloreado, tal y como muestra la Figura 4.3. El tamaño de cada círculo es proporcional al número de instancias que pertenecen a ese *cluster*. Para describir un *cluster* usaremos su centro (o *centroid*). Vemos que, al mover el ratón por encima de cada círculo, se muestra en pantalla la información del *centroid* correspondiente. Sus propiedades son el promedio de las propiedades de las instancias agrupadas en el *cluster*. Si además presionamos la tecla *mayúsculas*, aparecerá el botón situado bajo la tabla de propiedades del *centroid* que nos permite generar un nuevo dataset con las instancias agrupadas en ese *cluster*.

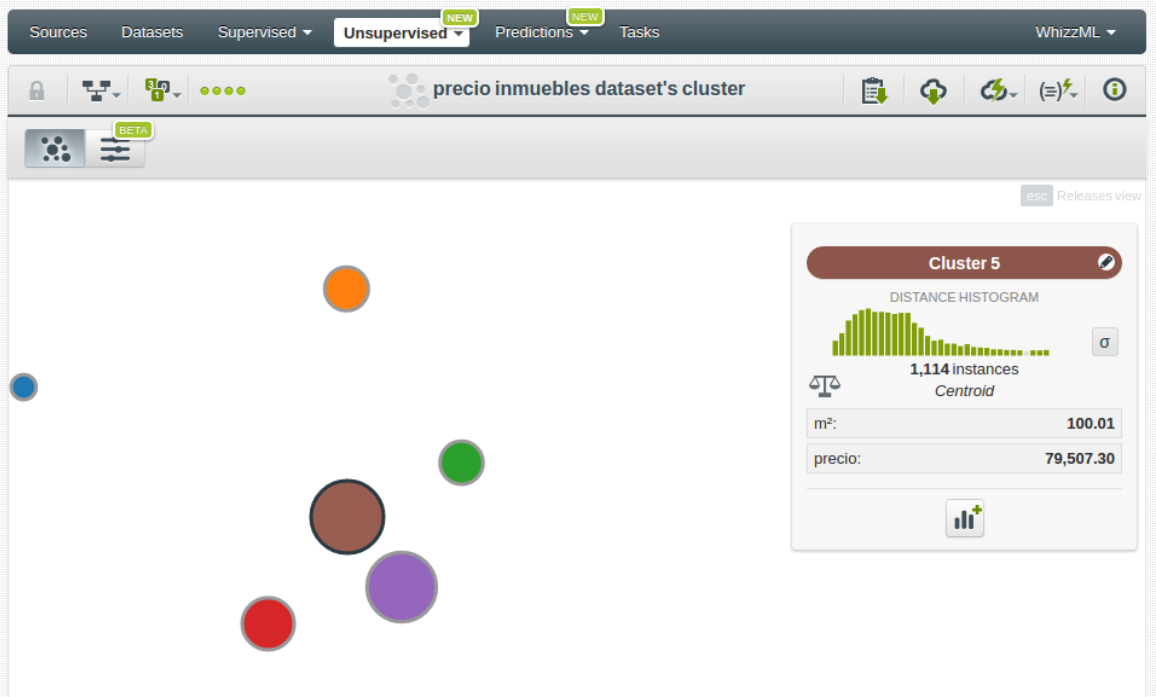


Figura 4.3: Visualización de los clusters creados con *g-means*. Cada círculo simboliza un grupo o *cluster*.

Otro elemento importante a tener en cuenta es la distancia existente entre los diversos *centroids*, ya que nos dará una idea de cuán distintos son los *clusters* generados. En nuestro gráfico, los círculos más alejados del *cluster* central representan los *clusters* con instancias más diferentes de las agrupadas en el *cluster* central y los más cercanos los más similares. Haciendo un clic sobre cualquier *cluster* del gráfico, éste pasará a ser el central y los demás se redistribuirán para mostrar sus distancias con relación a él. Las distancias entre los demás *clusters*, excluyendo el central, están adaptadas para evitar que los círculos se superpongan, y no tienen ninguna traducción en términos de las propiedades de dichos *clusters*.

En el ejemplo que hemos construido, vemos claramente que el *cluster* elegido como centro en este gráfico es el que se ha etiquetado como *Cluster 5*. Contiene 1114 instancias cuya distancia al *centroid* se encuentra distribuida según muestra el histograma. La lista de campos del centroid muestra los usados en el cálculo de la distancia: la superfície en metros cuadrados y el precio. La media de estos valores para todas las instancias que han sido agrupadas en este *cluster* es la que vemos asociada al *centroid*. El símbolo de la balanza que hay sobre la tabla de valores indica que estos valores se han escalado antes de calcular la distancia. Eso es especialmente importante en casos como éste, en que el campo precio tiene valores varios órdenes de magnitud mayores que el campo metros cuadrados.

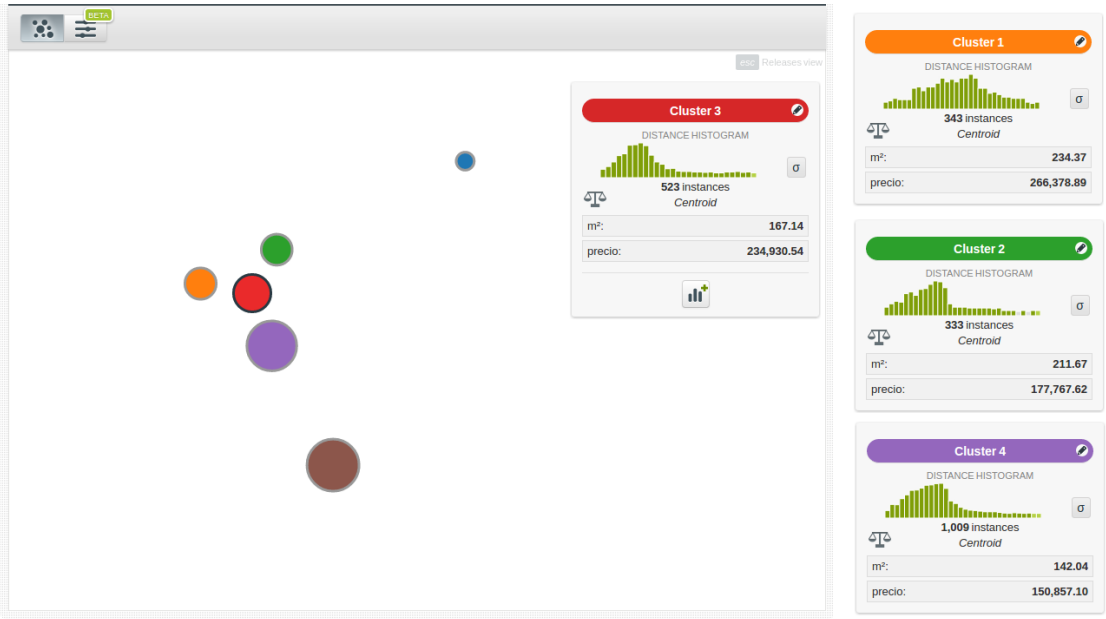


Figura 4.4: Visualización de los *clusters* creados con *g-means*: comparación de las características de los *clusters* más cercanos al central.

Si inspeccionamos otros *clusters* podremos ver qué tipo de instancias contienen y cuáles son las características que las agrupan. Por ejemplo, en el gráfico Figura 4.4 vemos que los *clusters* más cercanos al *cluster* rojo contienen inmuebles con superfícies y precios más similares. En cambio el *cluster* marrón y el azúl se diferencian más de ellos. Podemos analizar el detalle de los inmuebles que se han agrupado en el *cluster* azul creando un dataset que contenga esos datos. La Figura 4.5 muestra los histogramas de dicho dataset. Los 88 inmuebles que contiene son bastante más grandes de lo habitual, y su precio por tanto también está en lo más alto del rango.

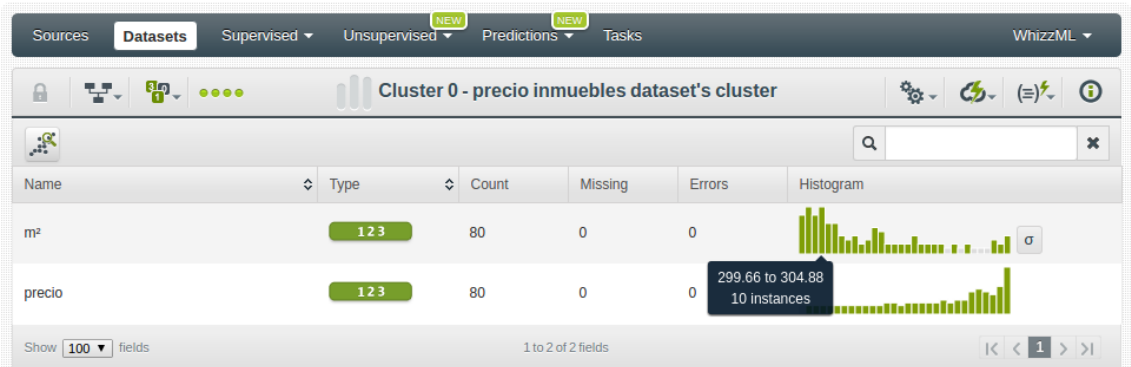


Figura 4.5: Visualización de los datos agrupados en un *cluster*.

Si queremos una vista más global de cómo se agrupan los datos podemos crear un *Batch centroid*. Esta acción está disponible en el menú de nuestro *cluster*. Seleccionando el *Dataset* que usamos originalmente para crear el *cluster*, el *Batch centroid* asigna el nombre del *cluster* al que pertenece cada fila. El resultado puede ser descargado en formato CSV o podemos generar un nuevo *Dataset* con una columna añadida que contendrá esta etiqueta. La Figura 4.6 nos muestra el resultado de crear un nuevo *Dataset* con un *Batch centroid*.

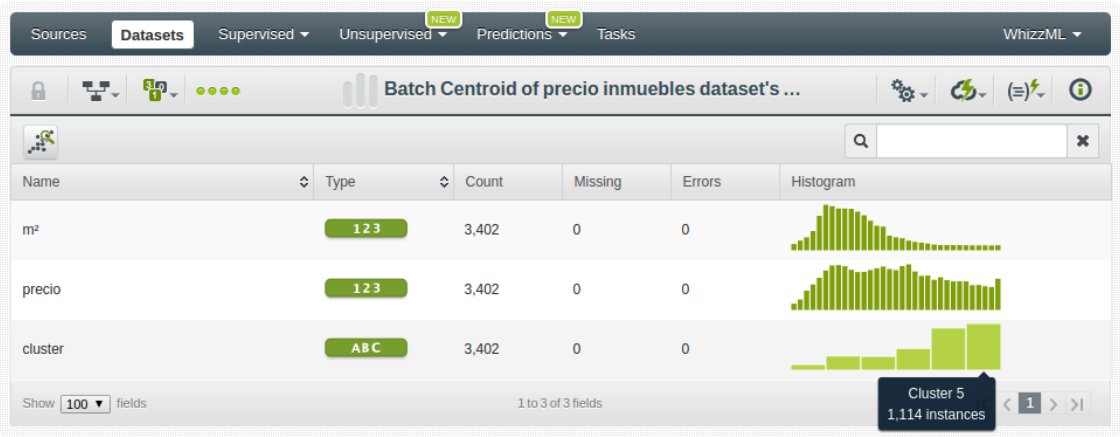


Figura 4.6: *Dataset* generado a partir de un *Batch centroid*. El *Batch centroid* asigna el nombre del *cluster* al que pertenece cada una de las filas del *Dataset* original.

Para visualizar mejor esta información, podemos usar el gráfico llamado *Dynamic scatterplot* usando el botón que se muestra en la Figura 4.7.

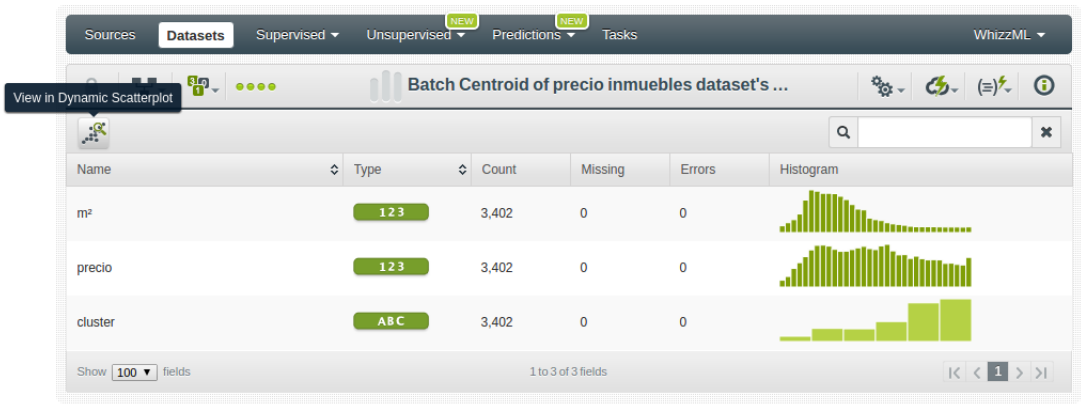


Figura 4.7: Acción que genera un *Dynamic scatterplot* desde un *Dataset*.

En él se representa un subconjunto de los puntos del *Dataset* elegidos al azar en un gráfico cartesiano de dos ejes. Podemos elegir qué campo queremos representar en cada eje y también podemos elegir otro campo para que los colores se asignen según su contenido. En nuestro caso, la Figura 4.8 usa los metros cuadrados y el precio como ejes y los colores se han asignado según el nombre del *cluster*. Se observa claramente las regiones definidas por los distintos *clusters* y cuál es la distribución de datos en esas regiones.

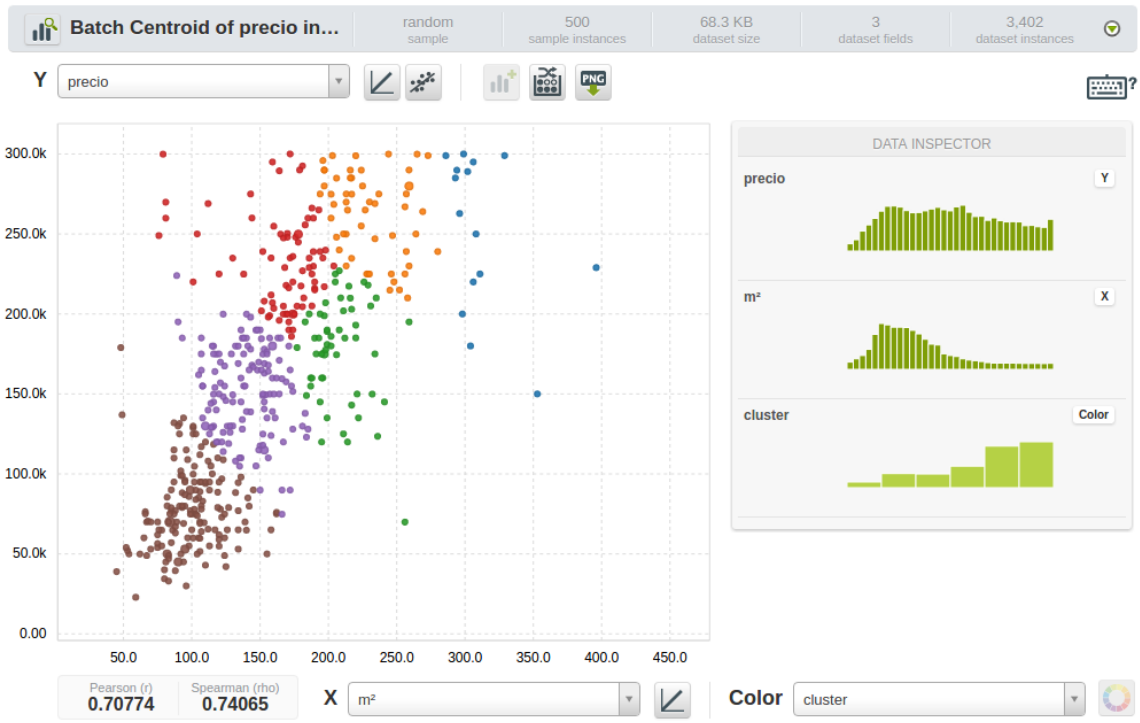


Figura 4.8: *Dynamic scatterplot* del *Dataset* generado por un *Batch centroid*. Los datos muestran un subconjunto de las filas de nuestro dataset coloreadas según el *Cluster* en qué han sido clasificadas.

Fijando el número de grupos: *k-means*

Si sabemos el número de grupos que queremos obtener, podemos dar esa información al crear el *clustering*. Ese número de clusters que queremos obtener será la k en nuestro algoritmo *k-means*.

En *k-means* cada fila de nuestro fichero corresponde a un punto en un espacio tantas dimensiones como atributos nuestro dataset. Inicialmente, el algoritmo elige k puntos al azar dentro de este espacio como puntos de referencia. Luego elige la primera instancia de nuestro dataset y calcula la distancia de ésta a cada uno de ellos. Finalmente, vincula esa instancia al punto de referencia más cercano. Sucesivamente, se repite el cálculo para todas las instancias del dataset. Con este proceso se consigue formar k grupos con las instancias del dataset según su cercanía a los tres puntos iniciales, pero todavía falta conseguir que estos grupos sean los más compactos entre sí y distintos de los demás.

Para ello, se calcula el punto central de todas las instancias pertenecientes a cada grupo. A partir de aquí, se repetirá el proceso anterior, pero eligiendo como nuevos puntos de referencia los tres centros, con lo que se calcularán las distancias de cada instancia a cada uno de los centros y se vincularán con el más cercano. Generalmente, este proceso hará que las instancias se vayan recolocando, formando nuevos grupos que tendrán nuevos centros. Cuando los grupos generados se mantengan estables (las instancias se mantengan en el mismo grupo durante varias iteraciones) habremos terminado el proceso y dichos grupos serán los clusters generados.

Para poner un ejemplo, usaremos un *Dataset* como el que vemos en la Figura 4.9, que contiene las puntuaciones que algunos usuarios han otorgado a un listado de marcas de cerveza.

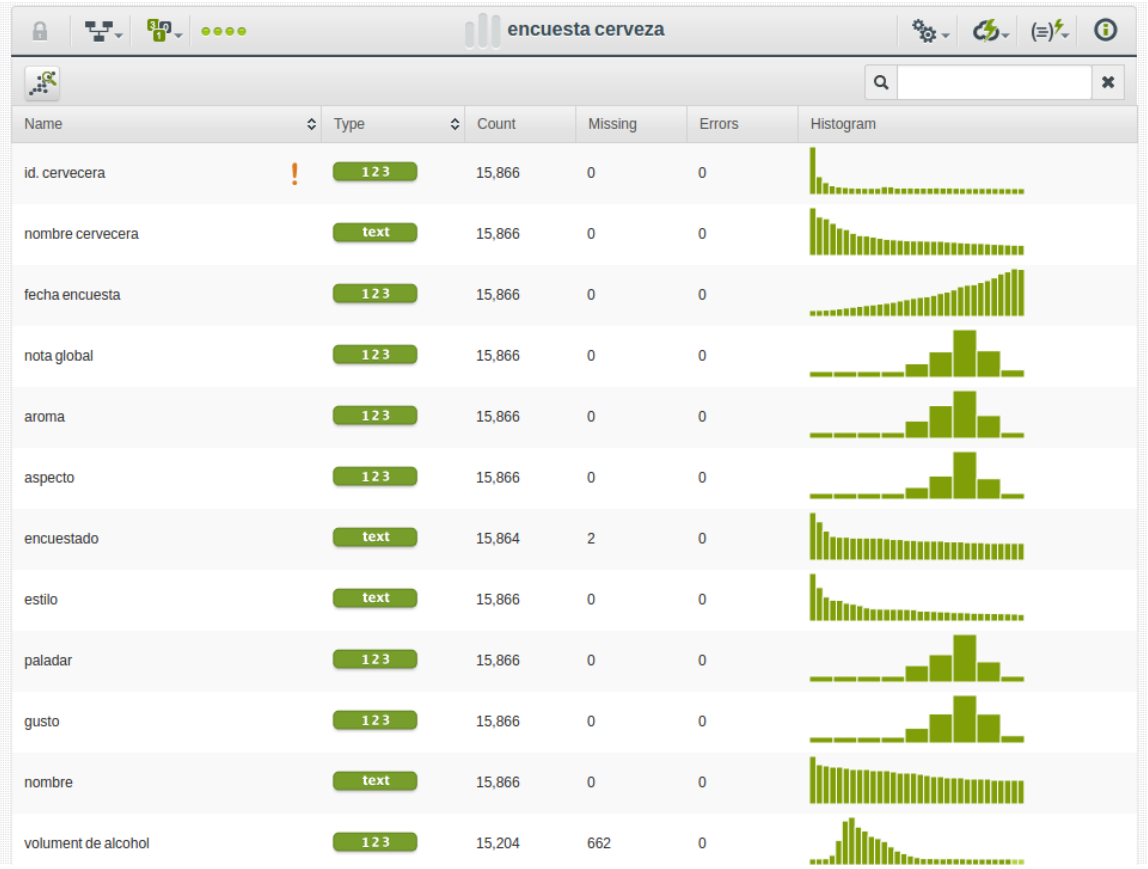


Figura 4.9: Dataset de cervezas: Cada instancia contiene las puntuaciones otorgadas por un usuarios a una marca de cerveza según su aroma, paladar, aspecto, gusto así como el volumen de alcohol que contiene.

Nuestro objetivo es clasificar dichas marcas en tres grupos para poder recomendar alternativas a nuestros clientes. Sabiendo el grupo al que pertenecen las cervezas que compran normalmente, podemos ofrecer otras marcas de características similares. Los datos con que contamos incluyen algunos datos que se usan como información de referencia, como el nombre de la cerveza, el de la compañía cervecera, o el identificador del usuario que puntúa. Los atributos que realmente califican la cerveza son la nota global, su aroma, gusto, aspecto, paladar y el volumen de alcohol. Este tipo de atributos son los que nos dicen si una cerveza es similar a otra o no y por lo tanto serán los que nos interesará usar en el clustering.

Figura 4.10: Configuración personalizada de un cluster.

Construiremos pues un modelo de *clustering* usando la pantalla de configuración de *Cluster*, tal como vemos en la Figura 4.10. Además de fijar el número de clusters a generar (k=3), también marcaremos el botón de modelado y listaremos los campos que hemos mencionado en el apartado anterior como campos de referencia. Estas opciones se muestran en la Figura 4.11, pero vamos a explicarlas un poco más para entender qué resultado perseguimos con ellas.

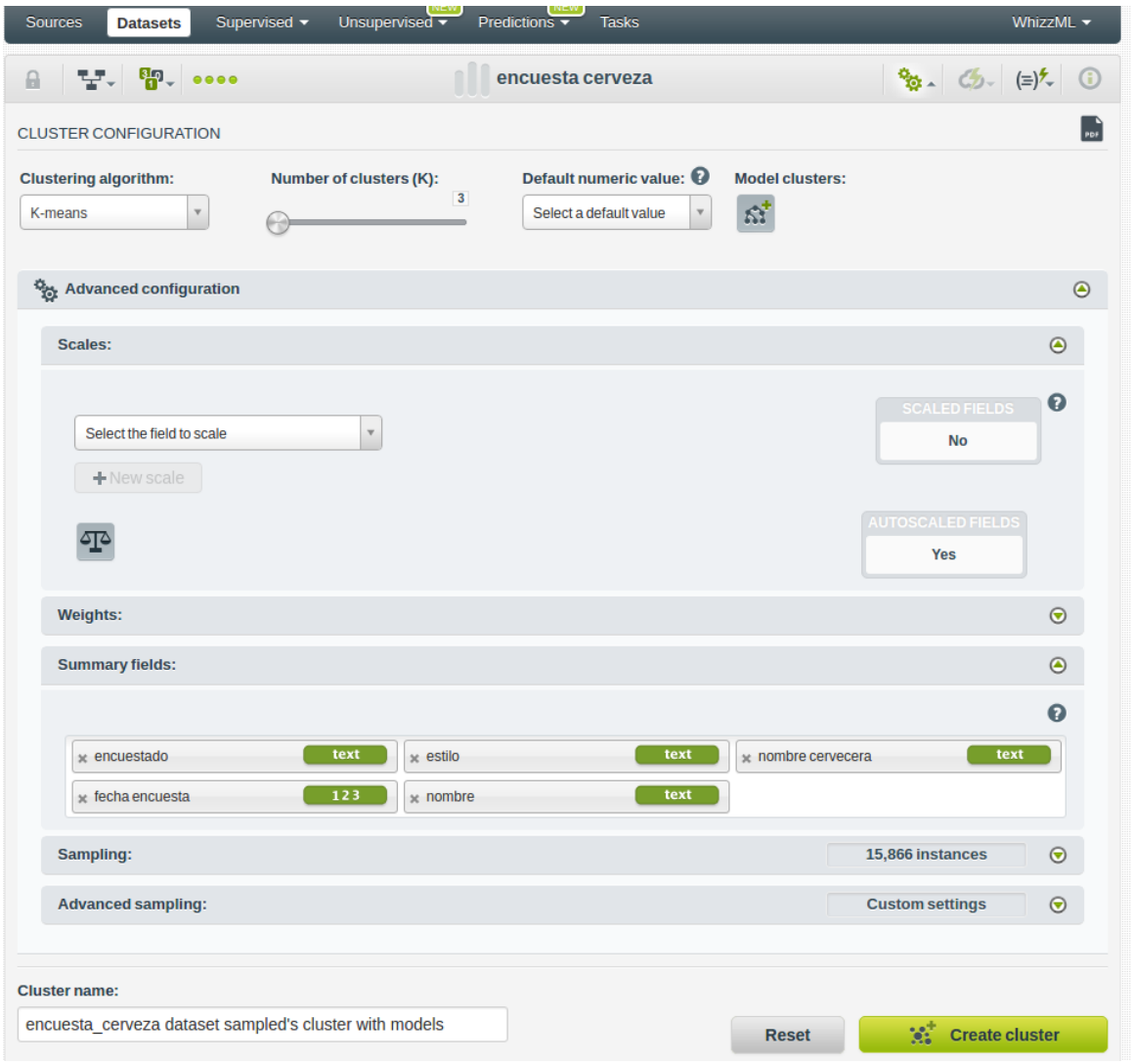


Figura 4.11: Pantalla de configuración de un *Cluster*: las opciones usadas son *k=3*, botón de modelado activado y un listado de campos de referencia (*summary fields*).

Los campos listados bajo el epígrafe *summary fields* son tenidos en cuenta solamente a nivel informativo, pero nunca son usados en el cálculo de la similitud entre los datos. Vimos ya, en el apartado anterior, que se puede crear un *Dataset* que contenga solamente los datos agrupados en un mismo *cluster*. Seguramente, en ese *Dataset* nos interesará tener la información de referencia, como el nombre de la cerveza analizada, y sin embargo no querremos que la diferencia de nombre sea tenida en cuenta cuando analizamos si las cervezas se parecen o no. Por eso, incluiremos el campo que contiene el nombre de la cerveza en los *summary fields* y, como éste, cualquier otro campo que contenga informaciones que queramos conservar pero que no influyen en la similitud de nuestras instancias.

El otro cambio de configuración es el botón de modelado. Al activarlo, se generará un árbol de decisión para cada *cluster*. El objetivo de dicho árbol de decisión es predecir si una instancia pertenece o no al *cluster* en cuestión. El modelo se construye usando el dataset original del que partimos con una columna más, donde se ha añadido la información sobre si la instancia pertenece al cluster o no.

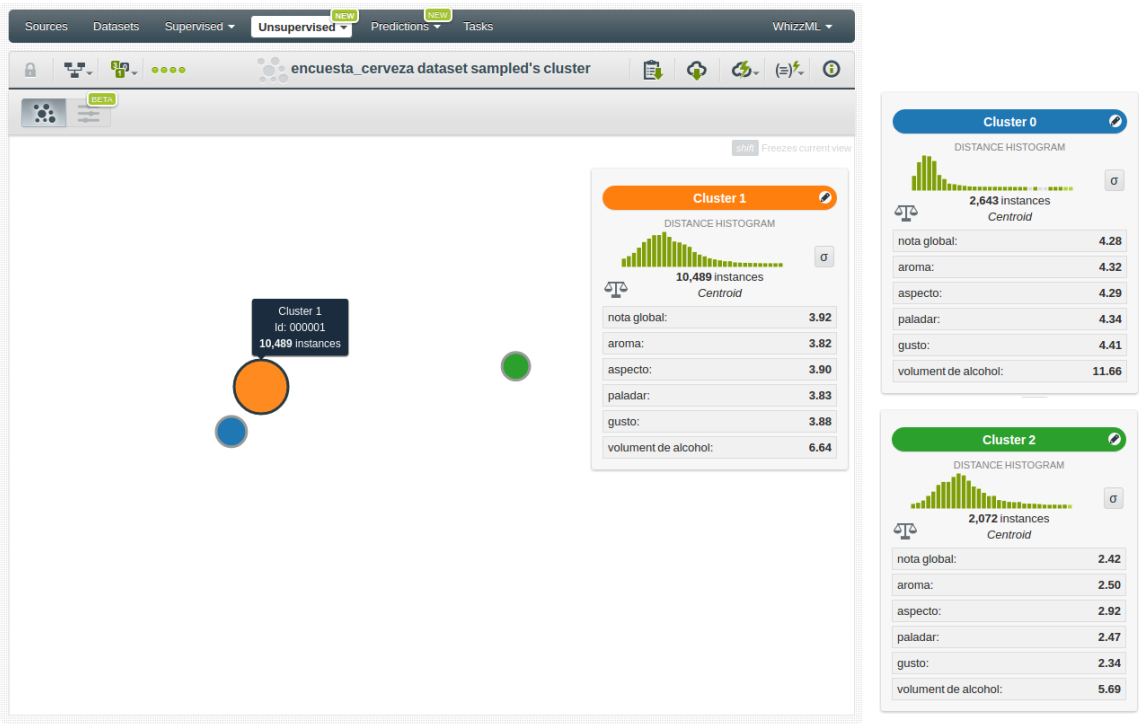


Figura 4.12: Visualización del *Cluster* para el *Dataset* de la encuesta sobre cervezas. Los *Centroids* presentan el promedio de las características de cada grupo.

Los *centroids* de los *clusters* obtenidos con este nuevo algoritmo se muestran en la Figura 4.12. Podemos ver un resumen de las distancias entre los puntos de un mismo *cluster* y también entre los distintos *centroids* usando el botón del *Cluster Summary Report*, tal como se muestra en Figura 4.13. Eso nos da una idea de como de compactos son los grupos y cuan distintos son entre sí.

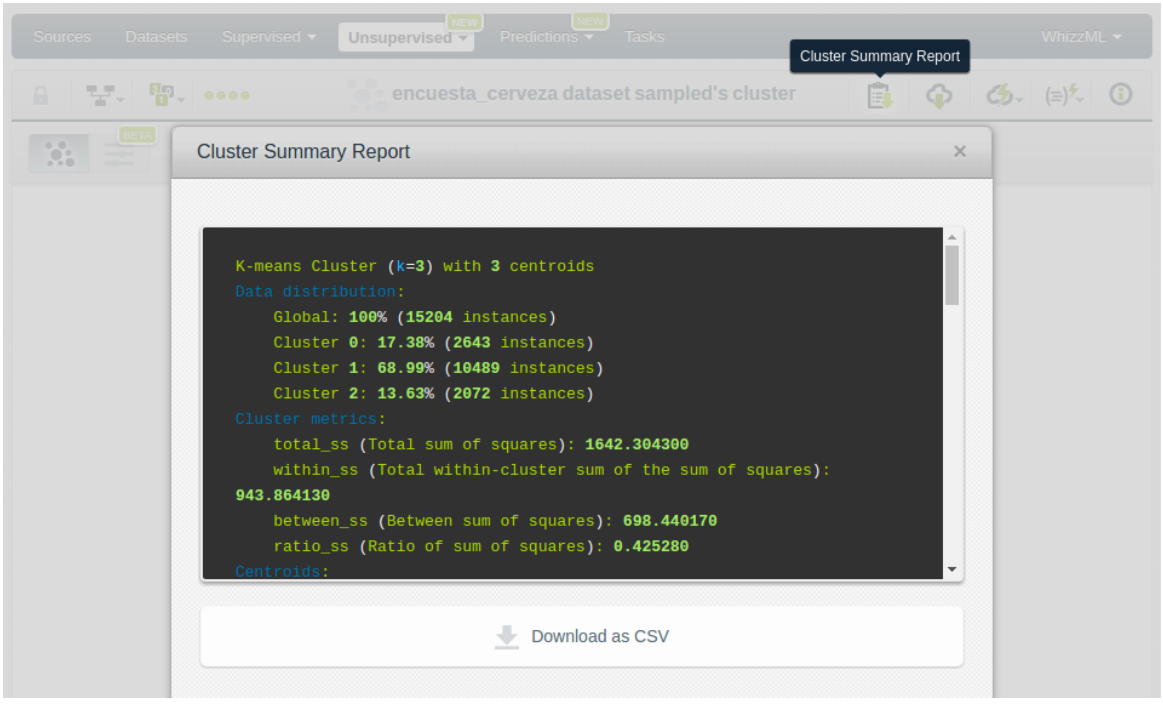


Figura 4.13: Resumen de las distancias entre los puntos pertenecientes a cada *cluster* y las distancias entre los *centroids*.

Aparentemente, la diferencia más evidente entre ellos es el volúmen de alcohol, pero para entender qué

factores hacen realmente que una instancia pertenezca a un cluster concreto podemos usar los modelos asociados que hemos creado al activar el botón correspondiente en la configuración del cluster (Figura 4.11). Para obtener el modelo correspondiente al *cluster 0*, deberemos apretar la tecla *mayúsculas* al pasar el ratón sobre él. Se mostrará el botón que vemos en la Figura 4.14, que nos permitirá acceder al modelo que predice la pertenencia a ese cluster.

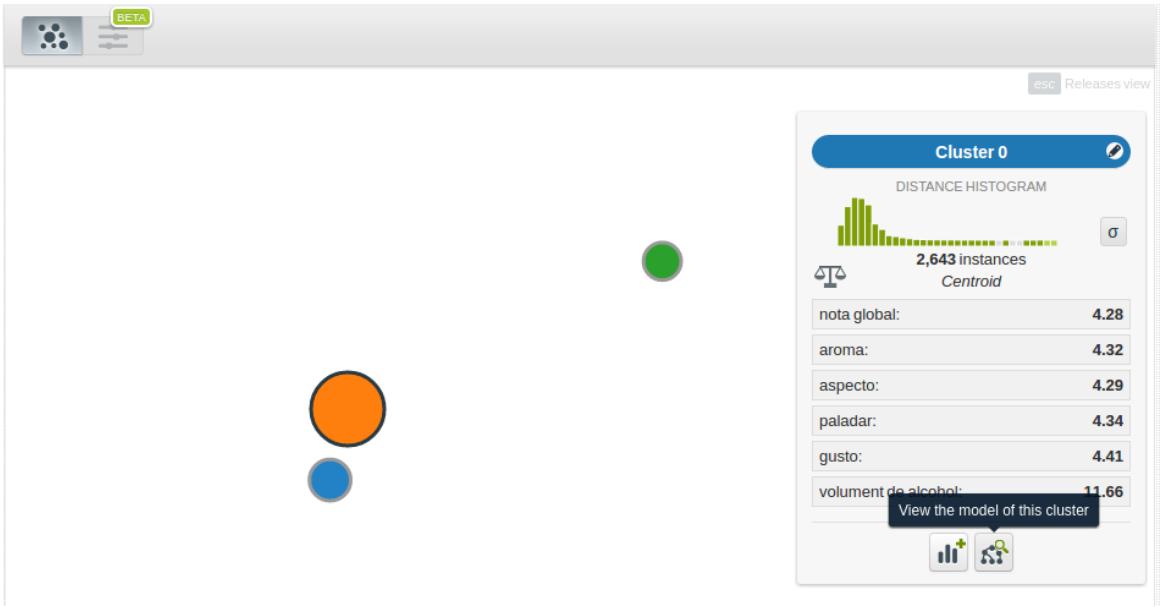


Figura 4.14: Botón de creación de modelos: Pulsando el botón crearemos el modelo que predice la pertenencia o no al *cluster 0*.

Si miramos la importancia de cada campo en ese modelo (Figura 4.16) vemos que el volumen de alcohol es precisamente el campo que más influye, seguido por el gusto y el paladar.

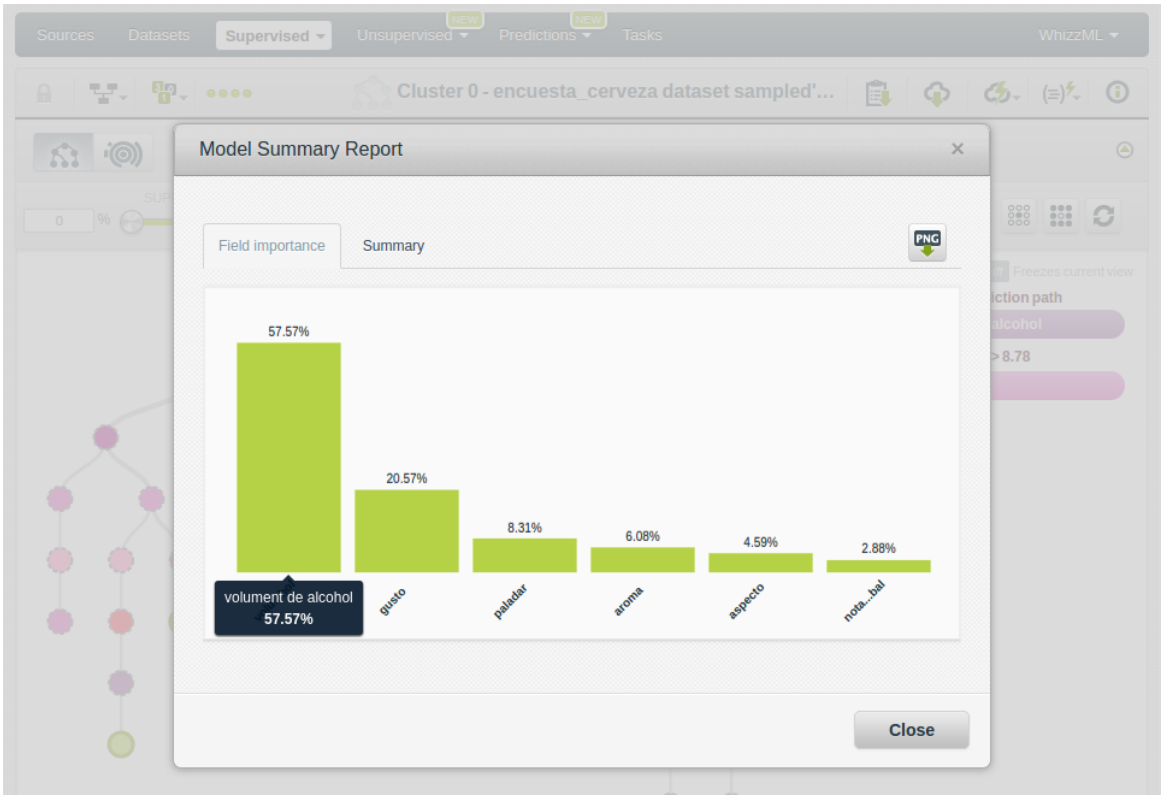


Figura 4.15: Importancia de los campos en el modelo de pertenencia al *cluster 0*.

Repitiendo el proceso que vimos en el ejemplo del dataset de inmuebles con que ilustramos el algoritmo de *g-means*, si creamos un dataset que almacene qué *cluster* se asigna a cada instancia y vemos su *scatterplot* eligiendo estos campos como ejes del gráfico, nos será fácil entender la distribución de las cervezas en cada grupo según sus propiedades (Figura 4.17). Según vemos, las cervezas más suaves tanto por gusto como por volúmen de alcohol se agrupan en un *cluster*, mientras que las que tienen más gusto se separan en dos dependiendo de si su volumen de alcohol es alto o bajo.

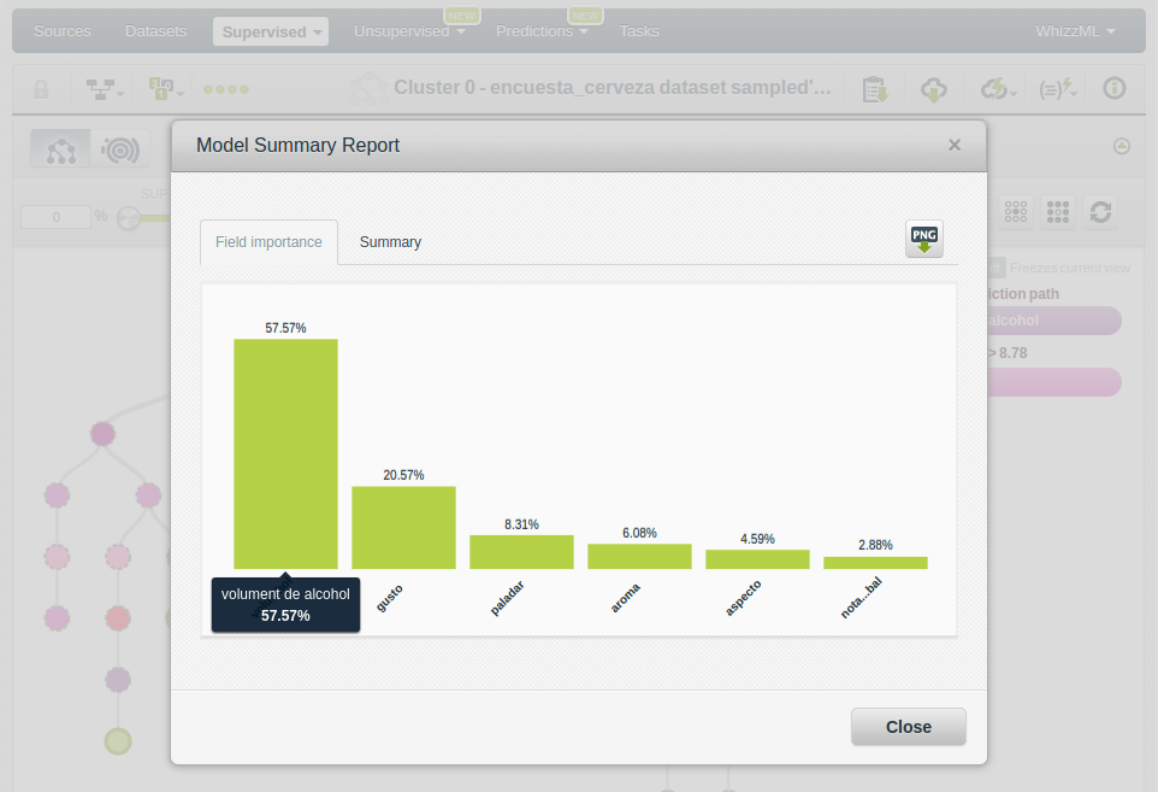


Figura 4.16: Importancia de los campos en el modelo de pertenencia al *cluster 0*.

Con estas informaciones, podemos escoger alternativas a las marcas habituales de nuestros clientes buscando primero el *cluster* en que se se clasifican y luego las otras marcas con características más parecidas dentro de este grupo.

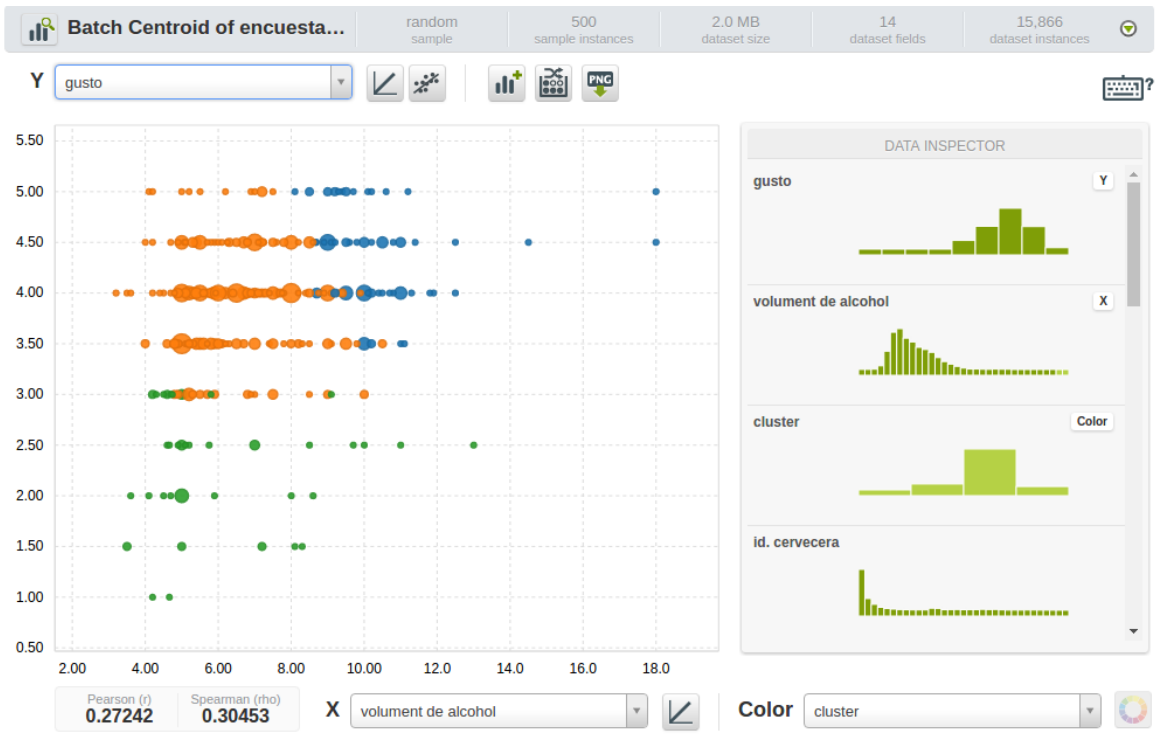


Figura 4.17: *Scatterplot* que representa la distribución de los tres *clusters* descubiertos en el *Dataset* de la encuesta sobre cervezas en función de su gusto y el volúmen de alcohol.

4.2. Detección de anomalías: Anomaly Detector

En los problemas de detección de anomalías el objetivo es encontrar aquellas instancias que se distinguen porque no siguen los patrones presentes en el resto de los datos. Para conseguir aislar estas instancias del resto, utilizaremos un detector de anomalías basado en un algoritmo conocido como *isolation forest*.

El algoritmo se basa en la idea de que si una instancia es muy anómala, como algunos de sus atributos serán muy distintos de lo habitual será fácil caracterizarla y separarla. Para hacerlo, el algoritmo construye un conjunto de árboles de decisión. Cada uno de ellos parte de un subconjunto distinto de los datos del *Dataset* escogido al azar. El objetivo de estos árboles es aislar cada instancia en un nodo distinto a base de establecer condiciones sucesivas sobre sus atributos. Estas condiciones van separando los grupos de instancias a medida que se baja en profundidad por el árbol. Las instancias que quedan a poca profundidad en el árbol se considera que son más anómalas, ya que se han podido diferenciar de las demás usando pocas condiciones. En cambio, las que quedan a mucha profundidad serán menos anómalas, ya que estas han requerido muchas más condiciones para diferenciarse de las demás al ser más parecidas. Basándose en esta observación, podemos asociar un *anomaly score* a cada instancia, comparando la profundidad a la que se separa en cada árbol con la profundidad media de los árboles construidos, tal como se muestra en la Figura 4.18.

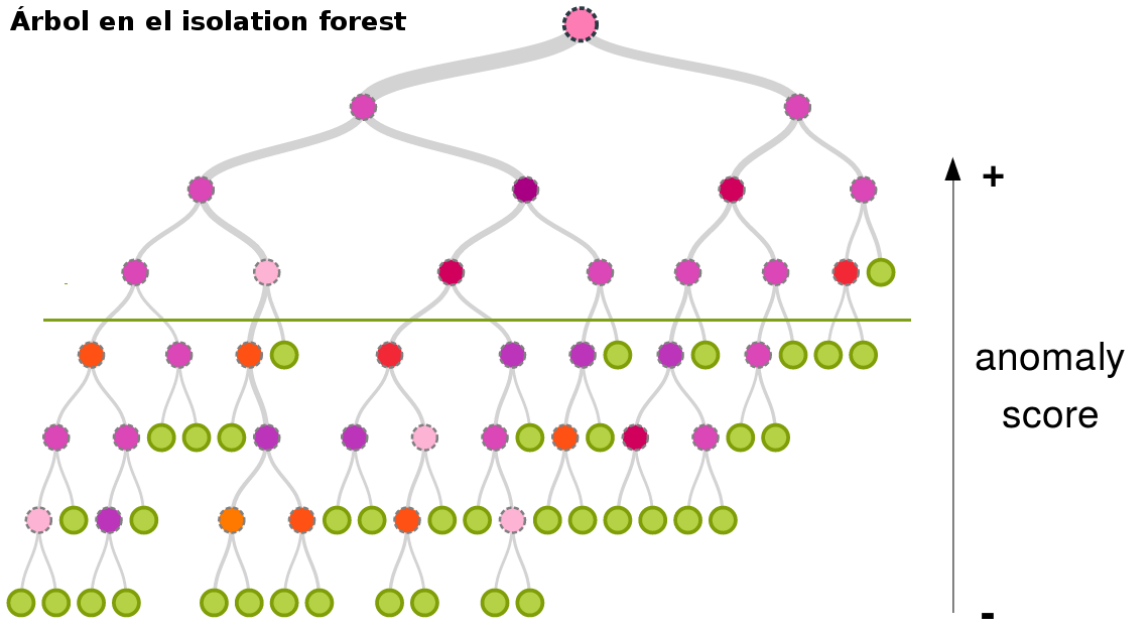


Figura 4.18: Esquema del cáculo del *anomaly score*. La línea horizontal marca la profundidad media de los árboles generados y las instancias se consideran más anómalas cuanto más arriba están los nodos en que se separan del resto.

El *anomaly score* es, por consiguiente, un número entre 0 y 1 que caracteriza el nivel de anomalía de cada instancia. El valor 0 indica total ausencia de anomalías y el 1 la máxima anomalía posible. En general, consideraremos que las instancia con un *anomaly score* mayor que 0,6 son anómalas, pero este límite puede variar según la distribución de nuestros datos.

Para ver cómo crear un detector de anomalías y qué informaciones podemos obtener con un detector de anomalías, usaremos un dataset donde tenemos algunos datos sobre préstamos. La información de que disponemos se muestra en la Figura 4.19 e incluye, entre otras variables, la cuantía del préstamo, los intereses, si hay impagos y la calificación del prestatario.

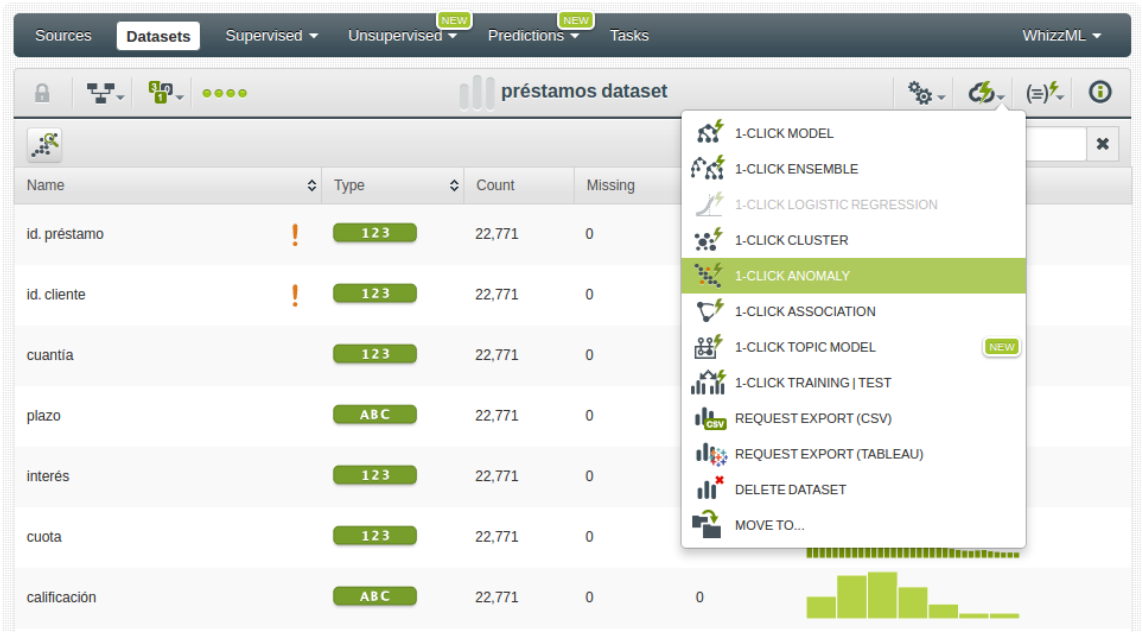


Figura 4.19: Creación de un detector de anomalías. Podemos crear un detector de anomalías en un clic desde la pantalla del *Dataset*. En el ejemplo, los atributos incluyen variables como la cuantía del préstamo, los intereses, los posibles impagos y la calificación del prestatario.

Partiendo de este *Dataset*, podemos crear un detector de anomalías mediante el enlace *1-click Anomaly*. De esa forma, se usarán los valores por defecto de los parámetros de configuración, como el número de árboles de decisión usados en el algoritmo y las n instancias más anómalas que se mostarán (por defecto las diez primeras). El resultado de esta acción se muestra en la Figura 4.20. A la izquierda de la pantalla encontramos la lista de instancias más anómalas.

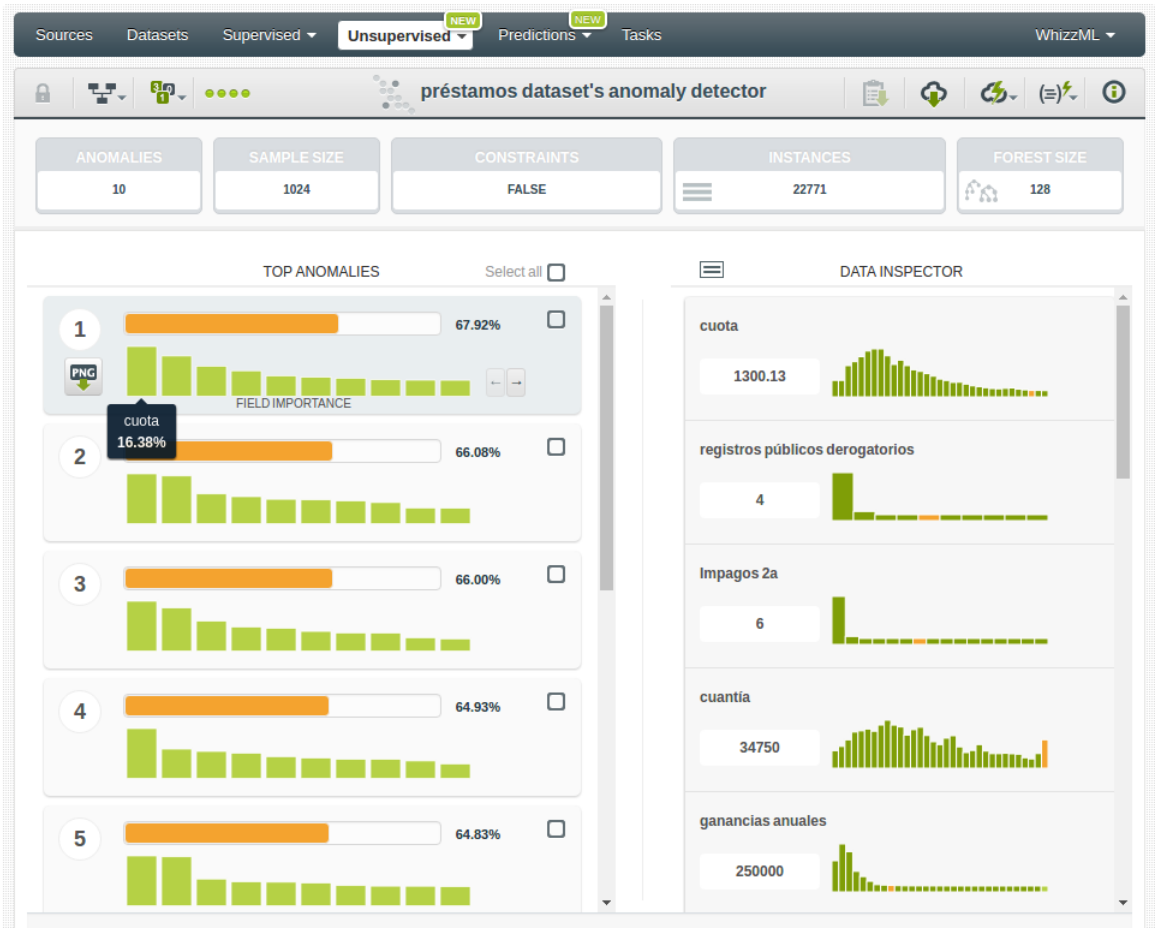


Figura 4.20: Visualización del detector de anomalías. A la izquierda se listan las 10 instancias más anómalas. A la derecha se muestran los campos que contribuyen a la anomalía de la instancia seleccionada y sus valores.

Fijémonos en la primera instancia de la lista. Su *anomaly score* es de 0,6792 y el gráfico de barras muestra la importancia que cada campo tiene en dicha anomalía. Cuando se selecciona una instancia con el ratón, se muestran a su derecha los valores de sus atributos. También se marca en naranja dónde caen esos valores dentro del histograma general formado por todos los valores del campo. Por ejemplo, vemos que el campo que contribuye más a marcar la primera instancia como anómala es la *cuota*. Su cuantía es de las más altas que se pueden encontrar en nuestros datos. Por eso la barra naranja que muestra su ubicación en el histograma del campo está muy a la derecha, coincidiendo con los valores más altos.

Si miramos el resto de valores de los campos de esta instancia, vemos que se trata de un préstamo con una cuota bastante alta, varios impagos en los últimos años, un importe alto y una calificación E, que es una de las peores. Este préstamo es altamente irregular, ya que el prestatario no cumple con las condiciones usuales para poder conseguir préstamos de esas caracterísicas. Las demás anomalías detectadas también presentan valores en sus campos que las hace inusuales desde el punto de vista de un experto financiero. No obstante, hay que recordar que el detector de anomalías las ha encontrado simplemente comparando sus valores con los más habituales y sin tener ningún conocimiento previo sobre las reglas usuales de los productos financieros. Por lo tanto, el modelo nos servirá para cualquier conjunto de datos, sin importar su naturaleza ni el dominio en el que se usen.

Tras detectar las anomalías de un *Dataset* tenemos varias opciones. Podemos crear un nuevo *Dataset* con ellas, para poder analizarlas mejor, o podemos extraerlas del *Dataset* original, para *limpiarlo* de casos anómalos. A menudo, esta eliminación de las instancias más anómalas de nuestro *Dataset* permite generar mejores modelos predictivos con los datos restantes. Para ello, podemos seleccionar las anomalías a incluir o excluir y usar el botón que hay al final de la pantalla para fijar la opción elegida al crear el nuevo *Dataset*.

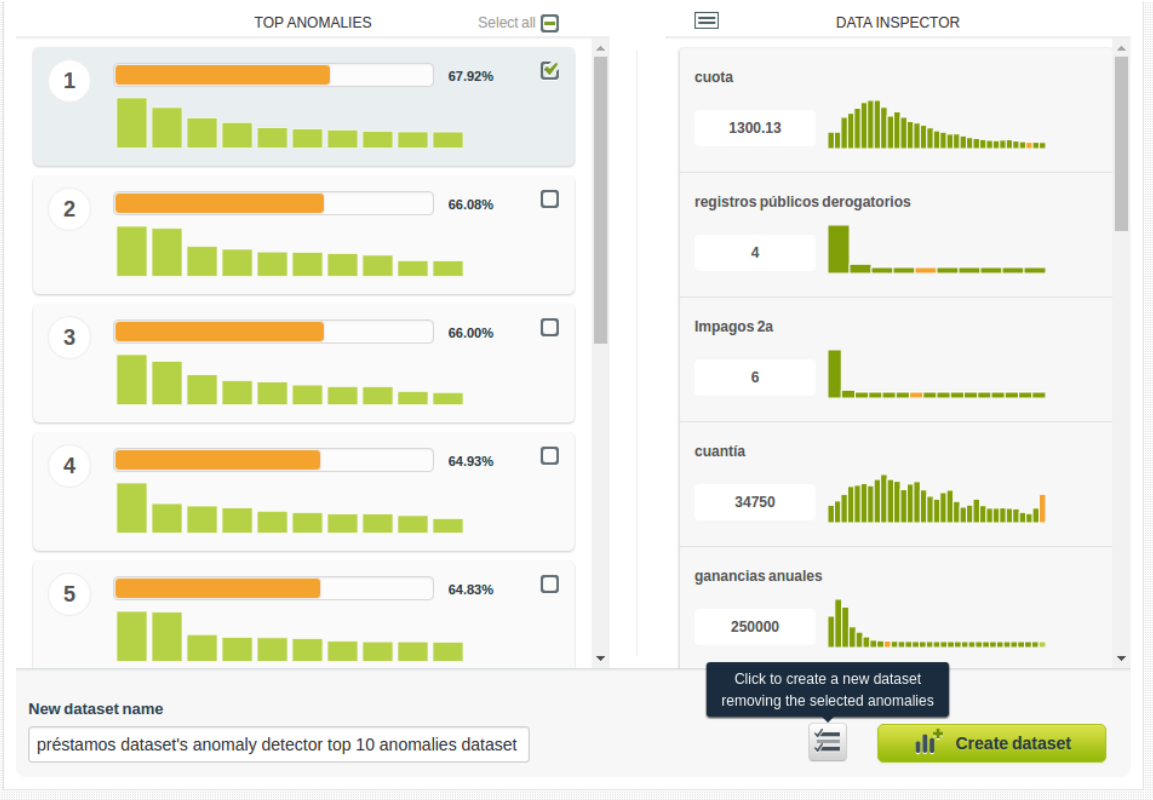


Figura 4.21: Creación de un nuevo *Dataset* donde se incluyen o excluyen las anomalías seleccionadas. En este caso, se generaría un *Dataset* con una única instancia seleccionada: la más anómala. Si, por el contrario, queremos excluirla del *Dataset* original, deberemos marcar el botón resaltado en la imagen.

Como hemos mencionado, el detector de anomalías construido por defecto muestra las diez instancias con mayor *anomaly score* encontradas en nuestro *Dataset*. No obstante, puede ocurrir que haya más o menos instancias anómalas. Para tener una visión global del cómo se distribuye el *anomaly score* en nuestros datos podemos crear un *batch anomaly score*, tal como se muestra en la Figura 4.22.

Figura 4.22: Creación de un *Batch anomaly score*: Se asignará un *anomaly score* a cada instancia usando el detector de anomalías.

Con este proceso, podremos obtener un nuevo *Dataset* donde, además de los datos originales, aparecerá una columna adicional que contendrá el *anomaly score* asignado por el detector de anomalías a cada fila. Analizando la distribución de valores de este nuevo campo, podremos saber si nuestros datos son más o menos homogéneos y, en función de eso, situar el valor del *anomaly score* que mejor separa las instancias anómalas. En la Figura 4.23 se muestra la distribución para nuestro ejemplo. Por la forma de la distribución, vemos que en este caso el valor límite de 0,6 es bastante ajustado, porque es donde empieza la cola de la gausiana.

Figura 4.23: *Batch anomaly score*: el *Dataset* generado mediante un *Batch anomaly score* contiene una última columna que almacena el *anomaly score* de cada una de las filas del *Dataset* original. La imagen muestra la distribución de sus valores.

Gracias a este histograma, podremos fijar el límite de *anomaly score* que nos parezca adecuado y filtrar los datos en función de él utilizando los métodos de filtrado de *Datasets* vistos en el *Módulo 2*.

4.3. Reglas de asociación: Association Rules

El último tipo de problema que nos plantearemos será el de encontrar posibles asociaciones entre valores de distintos campos que se dan a la vez con más frecuencia de la que cabría esperar. Estas asociaciones no son correlaciones, es decir, no buscamos campos que estén relacionados para todos sus valores. Tampoco es exactamente un problema de causalidad. Se buscan coocurrencias, es decir, diremos que hay una regla de asociación (*association rule*) cuando un predicado sobre un campo (por ejemplo, *campo1 = valor1*) ocurre a menudo a la vez que otro predicado (por ejemplo, *campo2 >valor2*). El ejemplo típico de este tipo de problema es el análisis de la cesta de la compra, donde se persigue saber qué productos se adquieren juntos para poder adecuar ofertas o su disposición en los expositores.

Hay que tener en cuenta que para que realmente haya una asociación relevante, la frecuencia con que coinciden ambos valores ha de ser mayor que la que se daría si se eligieran los dos independientemente por puro azar. Por eso, será necesario decidir cómo determinaremos si la coocurrencia observada es realmente significativa.

Usaremos un fichero de compras para ilustrar este tipo de modelos. El objetivo es pues encontrar relaciones como: *si se compra panecillos y hamburguesas, normalmente también se compra ketchup*. Como vemos, el tipo de reglas que expresarán estas asociaciones tienen el formato:

$$Antecedente \Rightarrow Consecuente$$

En el ejemplo, el antecedente sería *compra = panecillos y compra = hamburguesas* y el consecuente *compra = ketchup*. El antecedente puede contener más de un valor, mientras que en el consecuente sólo tendremos uno. Otros problemas en los que puede ser útil la detección de asociaciones es en los casos de analisis de fallos, detección de intrusos y biotecnología.

De entrada, vamos a analizar qué aspecto tendrán los datos que se usarán en este tipo de análisis. Al tratarse de una cesta de la compra, cada fila contendrá la información correspondiente a un tique de caja y en las columnas habrá el detalle del tipo de productos adquiridos. La estructura será pues similar a la de la Figura 4.24.

	A	B	C	D	E	F	G	H	I
1	cítricos	pan precocinado	margarina	sopas instantáneas					
2	fruta tropical	yogurt	café						
3	leche entera								
4	fruta de pepita	yogurt	crema de queso	paté de carne					
5	otros vegetales	leche entera	leche condensac	galletas					
6	leche entera	mantequilla	yogurt	arroz	limpiador abrasivo				
7	bollos								
8	otros vegetales	leche pasteuriza	bollos	cerveza embotel	vermut				
9	macetas								
10	leche entera	cereales							
11	fruta tropical	otros vegetales	pan blanco	agua embotellad	chocolate				
12	cítricos	fruta tropical	leche entera	mantequilla	cuajada	yogurt	harina	agua embotellad	platos
13	ternera								
14	frankfurt	bollos	soda						
15	pollo	fruta tropical							
16	mantequilla	azúcar	fruta/zumos	periódicos					
17	fruta/zumos								
18	fruta/vegetales empaquetados								
19	chocolate								

Figura 4.24: Estructura de los datos usados en el análisis de la cesta de la compra. Cada fila contiene información sobre un tique de caja. El número de columnas es variable y contiene los tipos de producto comprados.

Hay otros formatos alternativos igualmente válidos para el análisis. Podemos tener columnas con información de referencia, como el número de tique o la fecha de compra y todos los productos adquiridos acumulados en una sola columna y separados por un carácter distintivo. La Figura 4.25 muestra un ejemplo de este tipo de estructura.

	A	B
1	tique	productos
2	342533	cítricos,pan precocinado,margarina,sopas instantáneas
3	342534	fruta tropical,yogurt,café
4	342535	leche entera
5	342536	fruta de pepita,yogurt,crema de queso,paté de carne
6	342537	otros vegetales,leche entera,leche condensada,galletas
7	342538	leche entera,mantequilla,yogurt,arroz,limpiador abrasivo
8	342539	bollos
9	342540	otros vegetales,leche pasteurizada,bollos,cerveza embotellada,vermut
10	342541	macetas
11	342542	leche entera,cereales
12	342543	fruta tropical,otros vegetales,pan blanco,agua embotellada,chocolate
13	342544	cítricos,fruta tropical,leche entera,mantequilla,cuajada,yogurt,harina,agua embotellada,platos
14	342545	ternera
15	342546	frankfurt,bollos,soda
16	342547	pollo,fruta tropical
17	342548	mantequilla,azúcar,fruta/zumos,periódicos

Figura 4.25: Estructura de los datos usados en el análisis de la cesta de la compra. Cada fila contiene informaciones de referencia, como el número de tique o la fecha de compra, y un único campo contiene los tipos de producto comprados, que se han separado mediante un carácter distintivo.

Al subir los datos de la Figura 4.25, el objeto *Source* interpreta la columna que contiene los productos como un campo de tipo *Items*, con lo que analizará cada uno de los valores separados por comas como una unidad. Al crear el *Dataset* correspondiente podremos ver las frecuencias con que aparece cada uno de ellos.

Volviendo al ejemplo original de la Figura 4.24, el *Dataset* correspondiente muestra que el producto comprado con más frecuencia es la *leche entera*, seguido de *otros vegetales*, *bollos*, etc. El histograma correspondiente se muestra en la Figura 4.26 y nos ofrece la información estadística de la distribución de los productos en todas las compras, pero usando Machine Learning seremos capaces de descubrir cuáles se compran a la vez frecuentemente.

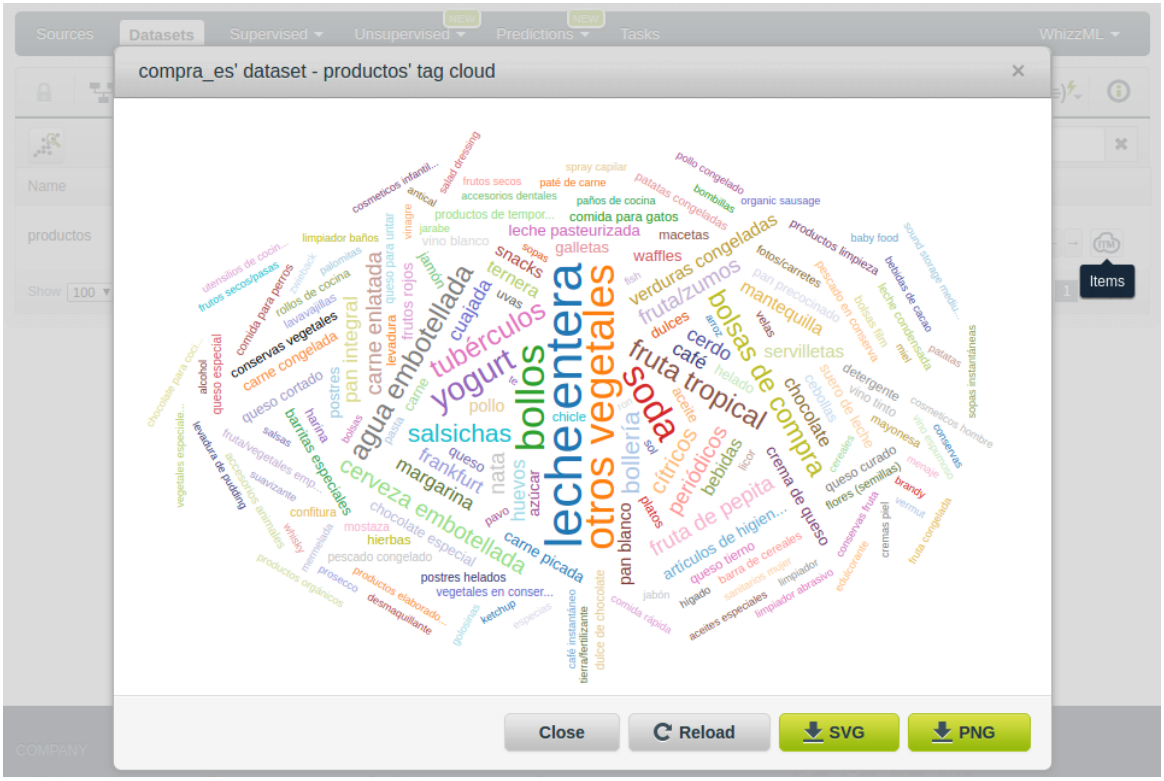


Figura 4.26: *Dataset* generado a partir del fichero de cestas de la compra. La nube de términos muestra los tipos de producto. El tamaño de letra es proporcional a la frecuencia con que se da cada término.

A partir del *Dataset*, podremos crear un detector de asociaciones usando la acción *1-click Association* tal como vemos en la Figura 4.27.

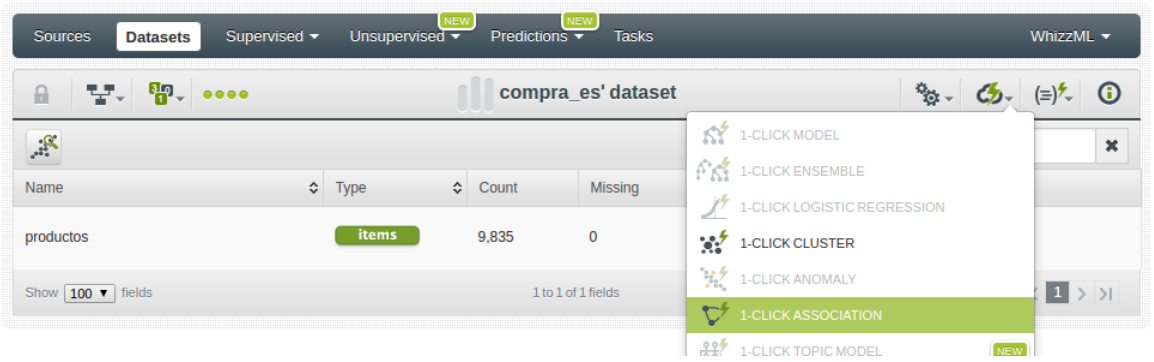
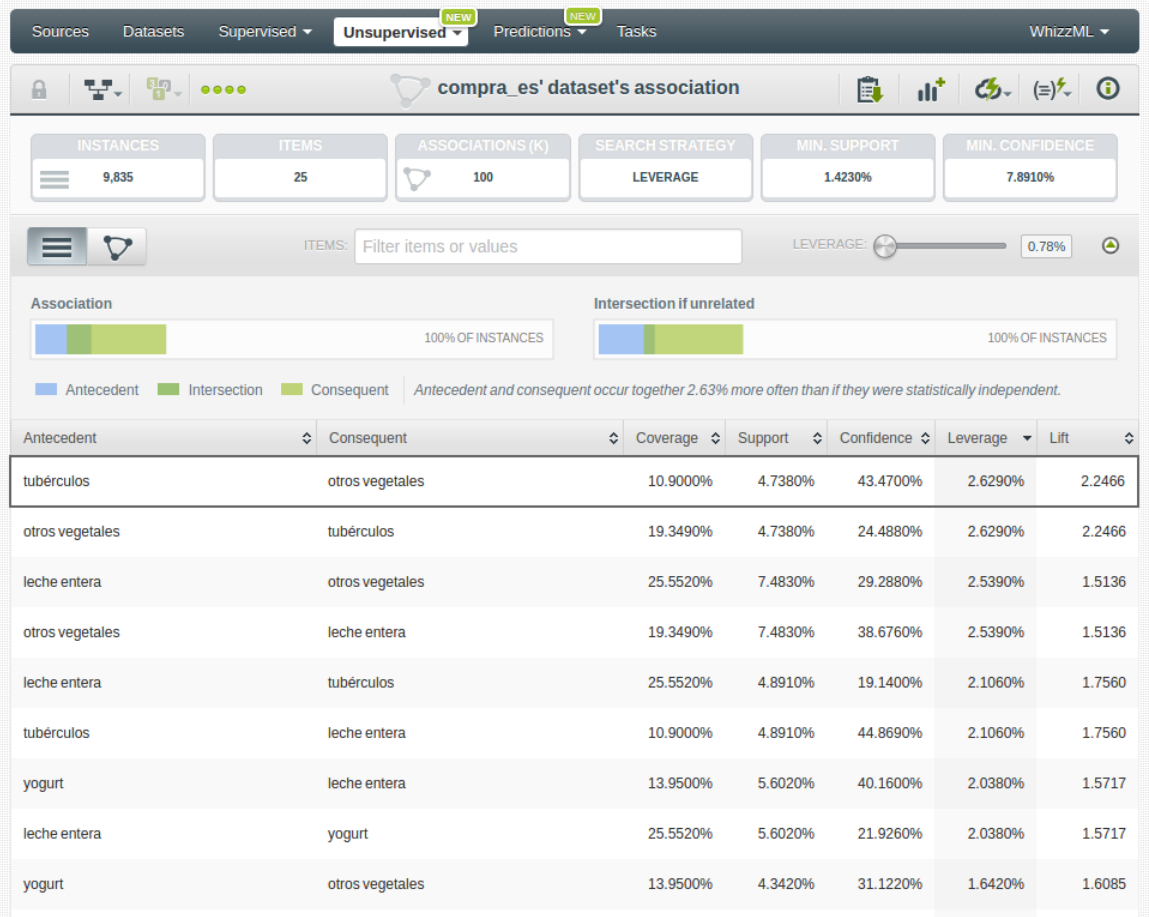


Figura 4.27: Creación de un detector de asociaciones en un clic desde un *Dataset*.

Como resultado, obtenemos una lista de 100 reglas que podemos ver en la Figura 4.28. La primera regla nos dice que cuando se compran *tubérculos* también se compran *otros vegetales*.



Antecedent	Consequent	Coverage	Support	Confidence	Leverage	Lift
tubérculos	otros vegetales	10.9000%	4.7380%	43.4700%	2.6290%	2.2466
otros vegetales	tubérculos	19.3490%	4.7380%	24.4880%	2.6290%	2.2466
leche entera	otros vegetales	25.5520%	7.4830%	29.2880%	2.5390%	1.5136
otros vegetales	leche entera	19.3490%	7.4830%	38.6760%	2.5390%	1.5136
leche entera	tubérculos	25.5520%	4.8910%	19.1400%	2.1060%	1.7560
tubérculos	leche entera	10.9000%	4.8910%	44.8690%	2.1060%	1.7560
yogurt	leche entera	13.9500%	5.6020%	40.1600%	2.0380%	1.5717
leche entera	yogurt	25.5520%	5.6020%	21.9260%	2.0380%	1.5717
yogurt	otros vegetales	13.9500%	4.3420%	31.1220%	1.6420%	1.6085

Figura 4.28: Reglas de asociación. La pantalla muestra las cien reglas de asociación que mejor apalancamiento proporcionan.

En la parte superior de la imagen vemos unas barras de colores azul y verde. Tal como explican las leyendas, la parte coloreada en azul representa las instancias que tienen el antecedente, en este caso las cestas donde se han comprado *tubérculos*. La parte coloreada en verde claro representa las instancias del consecuente, en el ejemplo son las cestas donde se han comprado *otros vegetales*. Vemos que hay una intersección coloreada en un verde azulado, que representa los casos en que ambos productos coinciden en una cesta. En el gráfico de la derecha la intersección corresponde a las cestas que contendrían ambos productos si éstas se creasen eligiendo productos al azar de entre todos los existentes. En cambio, en el gráfico de la izquierda la intersección muestra el número real de cestas que contienen ambos productos en la muestra de nuestros datos. Como podemos ver, el número de cestas que los contienen es mucho mayor que el que se daría por una elección al azar, por lo tanto podemos establecer que la compra de ambos productos está relacionada.

Hemos podido, pues, establecer que existe una asociación entre dos productos, pero sería interesante poder medir de algún modo en cuánto supera la intersección producida por la asociación a la que habría si esa asociación no existiese. Existen varias métricas que por si solas o combinadas nos permitirán cuantificar esa intensidad de la relación:

- **Cobertura:** Es el porcentaje sobre el total de casos en que se da el antecedente.
- **Soporte:** Es el porcentaje sobre el total de casos en que aparecen tanto el antecedente como el consecuente.
- **Confianza:** Es el porcentaje, sobre el total de casos que cumple el antecedente, donde se da el consecuente.
- **Apalancamiento (o *Leverage*):** Es la diferencia entre el soporte existente y el que se daría si antecedente y consecuente fuesen independientes.
- **Mejora (o *Lift*):** Es la proporción de casos en que antecedente y consecuente se dan juntos comparados con los que se darían si fuesen independientes.

La Figura 4.28 muestra los valores de dichas métricas para cada una de las reglas de asociación detectadas. Por tanto, sobre la primera regla de nuestro ejemplo podemos decir que:

- La cobertura es del 10,9 % porque los *tubérculos* aparecen en ese porcentaje de las compras.

- El soporte es del 4,738 % porque los *tubérculos* aparecen junto a los *otros vegetales* en ese porcentaje de cestas.

- La confianza es del 43,47 % porque los *otros vegetales* aparecen en ese porcentaje de las cestas que contienen *tubérculos*.

- El apalancamiento es del 2,629 % porque los *tubérculos* y los *otros vegetales* aparecen juntos 2,63 % más que si no estuviesen relacionados.

- La mejora es del 2,25 porque los *otros vegetales* aparecen 2,25 veces más en las cestas que tienen *tubérculos*.

Dependiendo de cuál sea nuestro objetivo, podemos usar cualquiera de estas métricas como criterio para seleccionar las n reglas de asociación que mejor puntúen en esa métrica. Por defecto, el detector de asociaciones selecciona las reglas que tienen mejor apalancamiento, pero este criterio se puede modificar usando la pantalla de configuración del detector de asociaciones de la Figura 4.29.

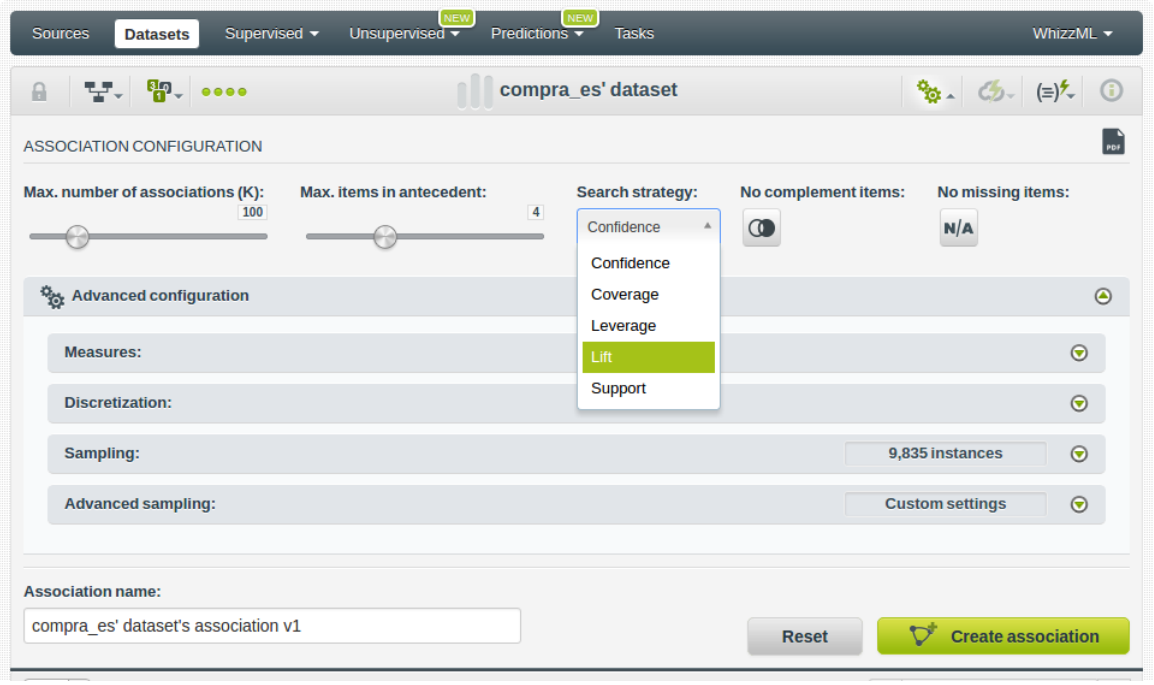
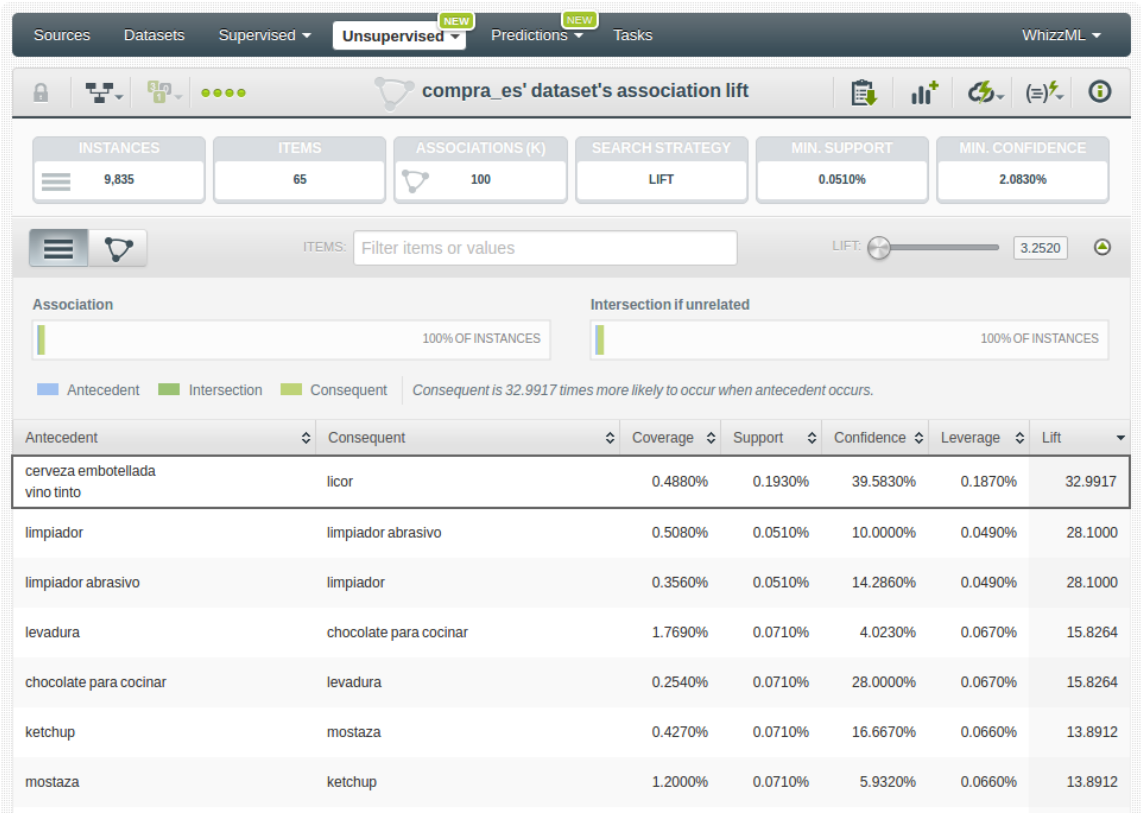


Figura 4.29: Pantalla de configuración del detector de asociaciones. Podemos decidir el número máximo de reglas seleccionadas y el criterio que guía su búsqueda se puede escoger entre cobertura, soporte, confianza, apalancamiento o mejora.

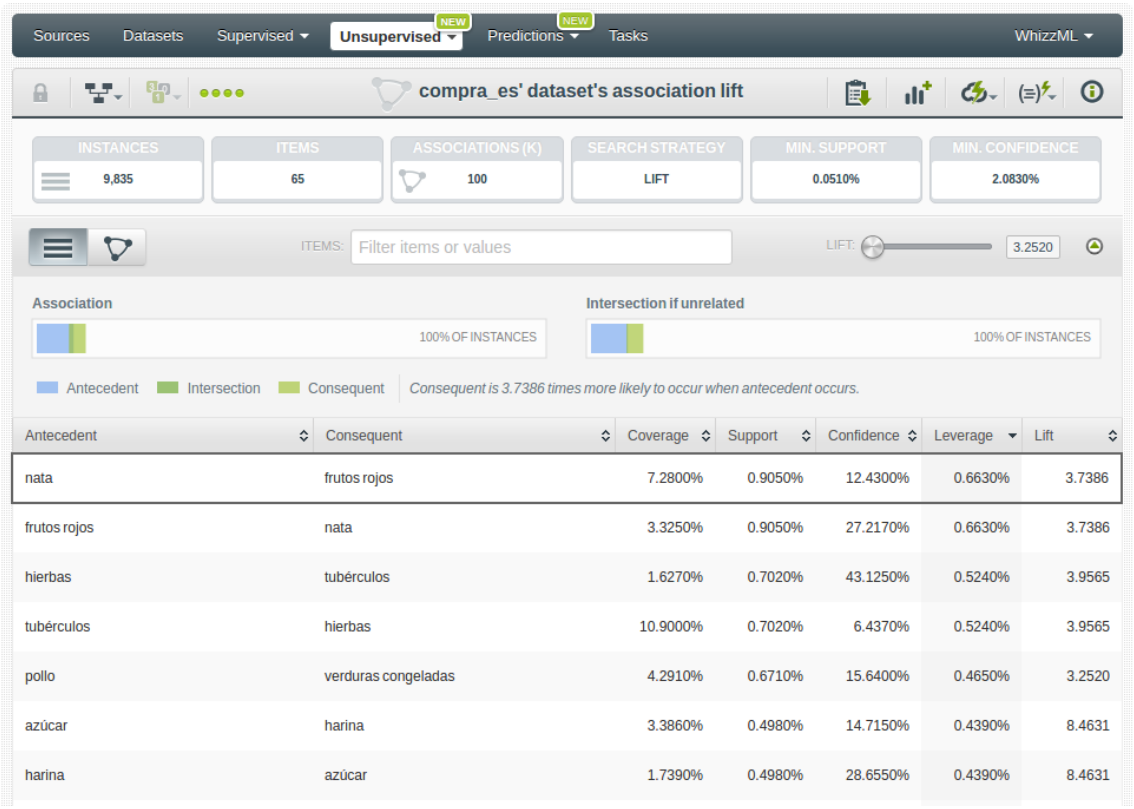
Si cambiamos el criterio de búsqueda del detector de asociaciones para usar la *mejora* en vez del *apalancamiento* (que es el criterio usado por defecto) recuperaremos otras reglas, que no ocurrirán tan a menudo en el total de compras, pero en que la vinculación entre antecedente y consecuente puede ser más fuerte. La Figura 4.30 muestra dichas reglas para nuestros datos de ejemplo. En la primera regla vemos como el *licor* aparece 33 veces más cuando la *cerveza embotellada* y el *vino tinto* están también en la cesta, aunque en realidad la cantidad de compras de estos productos es menor.



Antecedent	Consequent	Coverage	Support	Confidence	Leverage	Lift
cerveza embotellada vino tinto	licor	0.4880%	0.1930%	39.5830%	0.1870%	32.9917
limpiador	limpiador abrasivo	0.5080%	0.0510%	10.0000%	0.0490%	28.1000
limpiador abrasivo	limpiador	0.3560%	0.0510%	14.2860%	0.0490%	28.1000
levadura	chocolate para cocinar	1.7690%	0.0710%	4.0230%	0.0670%	15.8264
chocolate para cocinar	levadura	0.2540%	0.0710%	28.0000%	0.0670%	15.8264
ketchup	mostaza	0.4270%	0.0710%	16.6670%	0.0660%	13.8912
mostaza	ketchup	1.2000%	0.0710%	5.9320%	0.0660%	13.8912

Figura 4.30: Reglas de asociación. Lista de las reglas recuperadas con mayor mejora.

Si queremos encontrar las reglas detectadas que tienen más *apalancamiento* dentro de las que tienen más *mejora*, podemos cambiar la ordenación de la lista de reglas con un clic sobre la cabecera de la columna *apalancamiento*. El resultado se muestra en la Figura 4.31 y la primera regla nos dice que la *nata* y los *frutos rojos* aparecen juntos un 0.66 % más de lo que cabría esperar si se comprasen al azar. También podemos observar que las reglas no siempre son simétricas. Por ejemplo, vemos la regla que dice que si encontramos *pollo* en la cesta de la compra es probable que también encontremos *verduras congeladas*. En cambio, la regla inversa no aparece porque el hecho de tener *verduras congeladas* en la cesta de la compra no conlleva habitualmente que también se adquiera *pollo*.



Antecedent	Consequent	Coverage	Support	Confidence	Leverage	Lift
nata	frutos rojos	7.2800%	0.9050%	12.4300%	0.6630%	3.7386
frutos rojos	nata	3.3250%	0.9050%	27.2170%	0.6630%	3.7386
hierbas	tubérculos	1.6270%	0.7020%	43.1250%	0.5240%	3.9565
tubérculos	hierbas	10.9000%	0.7020%	6.4370%	0.5240%	3.9565
pollo	verduras congeladas	4.2910%	0.6710%	15.6400%	0.4650%	3.2520
azúcar	harina	3.3860%	0.4980%	14.7150%	0.4390%	8.4631
harina	azúcar	1.7390%	0.4980%	28.6550%	0.4390%	8.4631

Figura 4.31: Reglas de asociación. Lista de las reglas recuperadas con mayor mejora ordenadas por apalancamiento decreciente.

Además de esta lista exhaustiva de reglas, podemos ver una representación gráfica que nos ofrece una perspectiva global de las relaciones existentes entre los producto usando el botón que se muestra en la Figura 4.32. En este esquema, los círculos representan los productos que forman parte de las reglas. Las curvas que los unen simbolizan las asociaciones que detectadas entre los distintos productos (véase Figura 4.33).

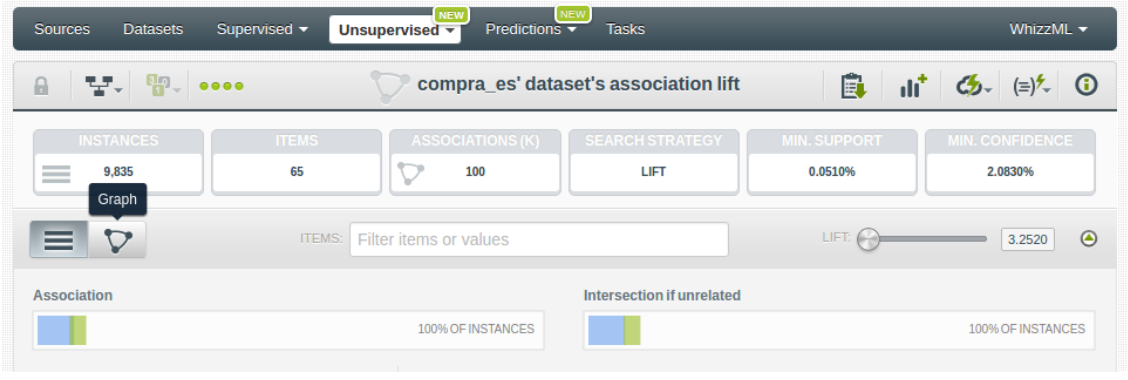


Figura 4.32: Botón de visualización gráfica de las reglas de asociación.

En ambas visualizaciones podemos filtrar las reglas, de manera que se muestren solo las que estén por encima de un cierto nivel de *mejora* (o el criterio que se haya elegido en la búsqueda en su lugar). Al elevar el umbral de *mejora* solo se mostrarán las asociaciones más fuertes. En el gráfico, el grosor de las líneas es un indicador de la magnitud de la *mejora* de la regla en cuestión. Para nuestro ejemplo, la regla con más *mejora* es la que relaciona la *cerveza embotellada*, el *vino tinto* y los *licores*, como ya vimos en el listado.

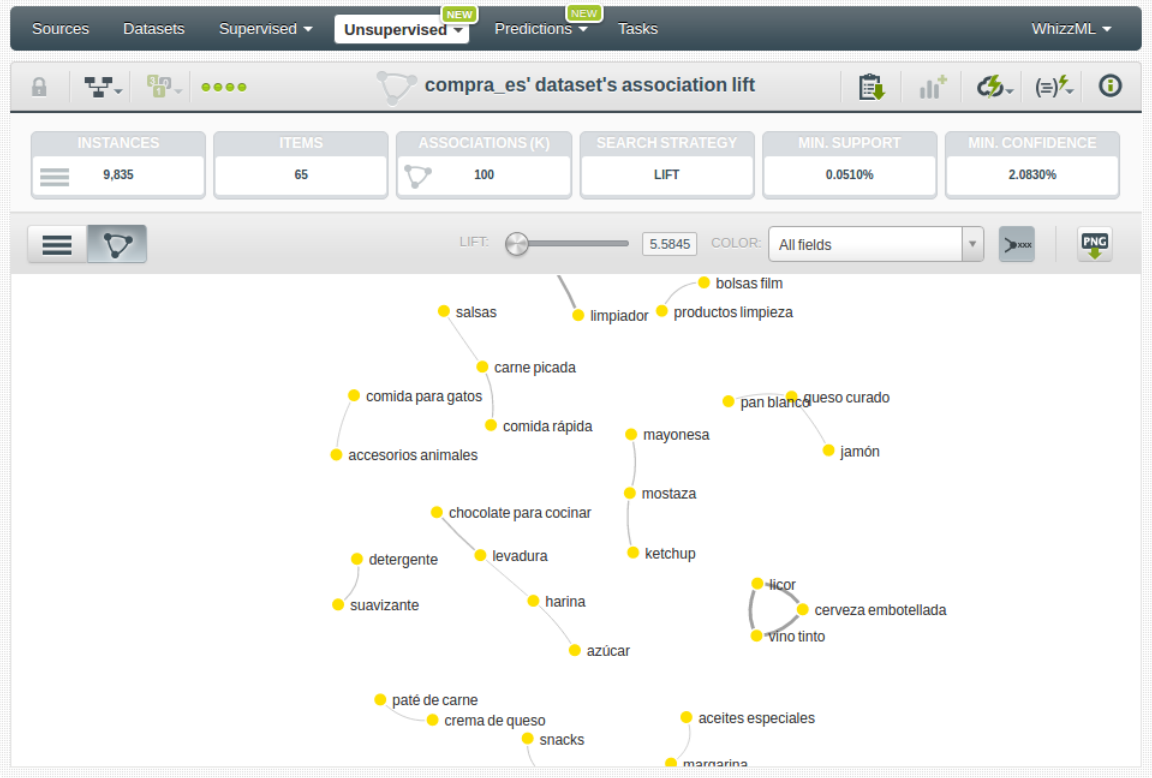


Figura 4.33: Visualización de las reglas de asociación en forma de grafo. Los círculos corresponden a los antecedentes y consecuentes y las líneas que los unen representan las asociaciones entre ellos.

En función de los resultados que queramos obtener, usaremos las reglas con más *apalancamiento* o *mejora*. Si usamos las primeras, estaremos detectando reglas que se dan con mayor frecuencia, mientras que si usamos las segundas detectaremos asociaciones más fuertes. Podemos igualmente usar cualquiera de las otras métricas en nuestra búsqueda de asociaciones, pero éstas dos son las más recomendadas, ya que normalmente caracterizan las asociaciones más relevantes.