ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
Grado en Ingeniería del Software

Minería de datos con Oracle Data Miner

Data mining with Oracle Data Miner

Realizado por
Benjamín Fernández Ruiz
Tutorizado por
Manuel Enciso García-Oliveros
Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, Enero 2016

Fecha defensa:
El Secretario del Tribunal

Resumen:

La intención del proyecto es mostrar las diferentes características que ofrece Oracle en el campo de la minería de datos, con la finalidad de saber si puede ser una plataforma apta para la investigación y la educación en la universidad.

En la primera parte del proyecto se estudia la aplicación “Oracle Data Miner” y como, mediante un flujo de trabajo visual e intuitivo, pueden aplicarse las distintas técnicas de minería (clasificación, regresión, clustering y asociación). Para mostrar la ejecución de estas técnicas se han usado dataset procedentes de la universidad de Irvine. Con ello se ha conseguido observar el comportamiento de los distintos algoritmos en situaciones reales.
Para cada técnica se expone como evaluar su fiabilidad y como interpretar los resultados que se obtienen a partir de su aplicación. También se muestra la aplicación de las técnicas mediante el uso del lenguaje PL/SQL. Gracias a ello podemos integrar la minería de datos en nuestras aplicaciones de manera sencilla.

En la segunda parte del proyecto, se ha elaborado un prototipo de una aplicación que utiliza la minería de datos, en concreto la clasificación para obtener el diagnóstico y la probabilidad de que un tumor de mama sea maligno o benigno, a partir de los resultados de una citología.

Palabras clave: Minería, Datos, Oracle, Clasificación, Regresión, Cluster, Asociación.

Abstract:

The intention of the project is to show the different features that Oracle provides in the field of data mining, in order to know if he can be a fit for research and education at the university platform.

In the first part of the project "Oracle Data Miner" application is studied and how, by a flow of visual and intuitive work, the different mining techniques (classification, regression, clustering and association) may apply. To display the execution of these techniques they have been used dataset from the University of Irvine. This has been achieved observe the behavior of different algorithms in real situations.
For each technique exposed to assess its reliability and how to interpret the results obtained from its implementation. Applying the techniques it is also shown using the PL / SQL. Thanks to that we can integrate into our data mining applications easily.

In the second part of the project, it has developed a prototype of an application that uses data mining, namely the classification for the diagnosis and the likelihood that a breast tumor is malignant or benign, from results cytology.

Keywords: Miner, Data, Oracle, Classification, Regression, Clustering, Association.

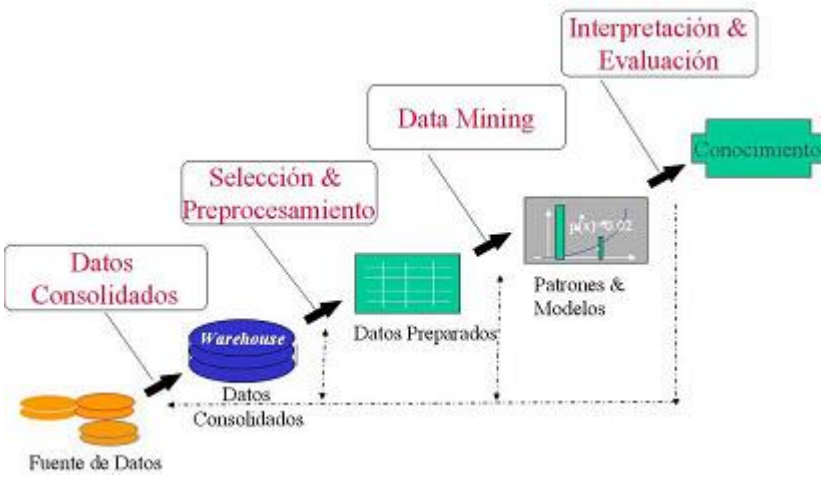
Índice

1. Introducción

1.1. ¿Qué es la minería de datos?

La minería de datos es un campo de las ciencias de la computación cuyo objetivo es encontrar patrones en grande conjuntos de datos. Para ello, utiliza los métodos de inteligencia artificial, aprendizaje automático, estadística y sistemas de bases de datos.

La acción de realizar minería de datos la podemos dividir en las siguientes fases:



Partimos de un conjunto de datos, los cuáles constituirán nuestro objeto de estudio. La fase inicial en la minería consiste en adecuar esta información, de manera que al aplicar los algoritmos, los resultados sean los más óptimos posibles. Por ejemplo, se normalizan datos, se convierten intervalos de valores continuos en atributos cualitativos, etc. Una vez preparado los datos iniciamos el proceso de minería. Este proceso principalmente está dividido en dos partes: la primera, entrenamiento y creación de un modelo y la segunda, aplicación del modelo a nuevos datos.

Una vez aplicado el modelo a los nuevos datos, procederemos a evaluar los resultados obtenidos.

1.2. ¿Qué queremos hacer y por qué?

La intención del proyecto es mostrar las diferentes características que ofrece Oracle en el campo de la minería de datos, con la finalidad de saber si puede ser una plataforma apta para la investigación y la educación en la universidad.

En la primera parte del proyecto se estudia la aplicación “Oracle Data Miner” y como, mediante un flujo de trabajo visual e intuitivo, pueden aplicarse las distintas técnicas de minería (clasificación, regresión, clustering y asociación). Para mostrar la ejecución de estas técnicas se han usado dataset procedentes de la universidad de Irvine. Con ello se ha conseguido observar el comportamiento de los distintos algoritmos en situaciones reales.
Para cada técnica se expone como evaluar su fiabilidad y como interpretar los resultados que se obtienen a partir de su aplicación. También se muestra la aplicación de las técnicas mediante el uso del lenguaje PL/SQL. Gracias a ello podemos integrar la minería de datos en nuestras aplicaciones de manera sencilla.

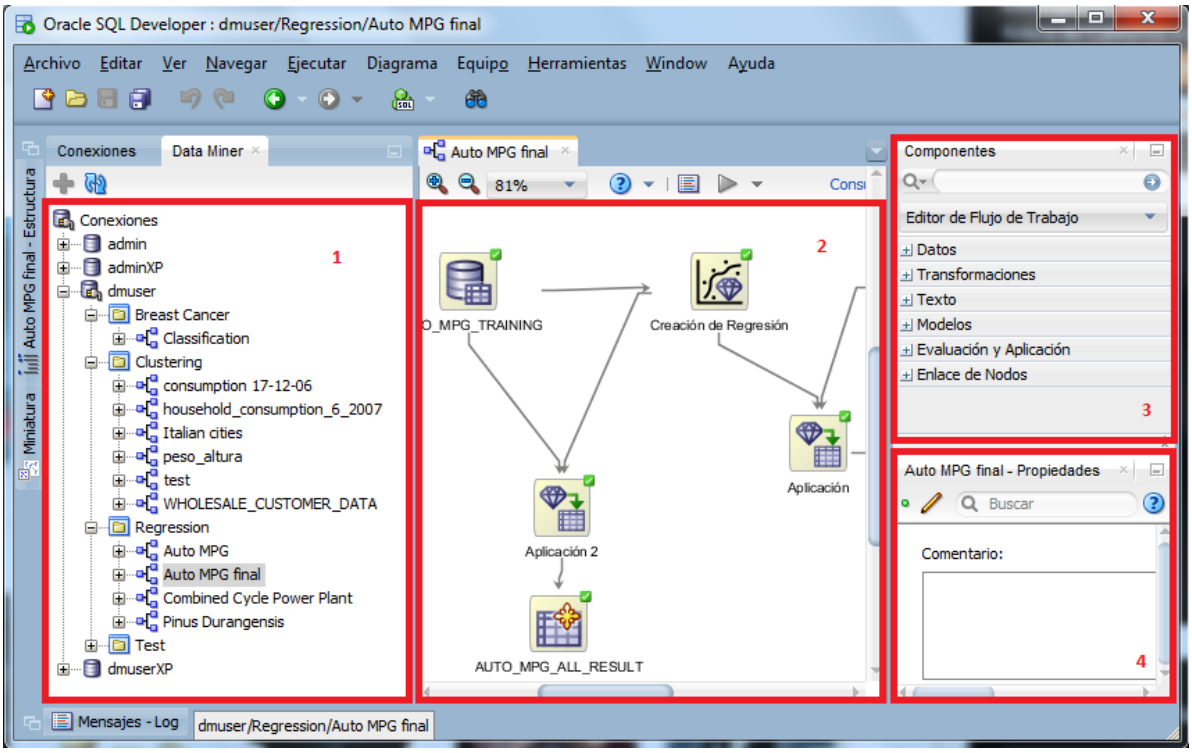
En la segunda parte del proyecto, se ha elaborado un prototipo de una aplicación que utiliza la minería de datos, en concreto la clasificación para obtener el diagnóstico y la probabilidad de que un tumor de mama sea maligno o benigno, a partir de los resultados de una citología.

2. Oracle Data Miner

La aplicación está dividida en cuatro partes muy bien definidas. En primer lugar, se encuentra el panel de conexiones (1), donde al realizar la conexión mediante nuestro usuario para la minería de datos, podemos observar los proyectos creados y en el interior de estos, los flujos de trabajo. Estos flujos de trabajo contienen la estructura de nodos que nos permite realizar la minería de forma visual e interactiva. Si abrimos un flujo de trabajo, este nos aparece en el área de trabajo (2). Es en esta área donde podemos añadir nodos y enlazarlos entre sí para lograr los objetivos deseados.

En el lado derecho del área de trabajo se sitúa el panel de componentes (3), donde se encuentran los diferentes tipos de nodos que podemos usar en nuestro flujo de trabajo.

Finamente debajo del menú de componentes, se sitúa un área (4) donde se muestran las diferentes opciones del nodo que tengamos seleccionado.



Los componentes (panel 3) que podemos usar en nuestro proceso de minería de datos son los siguientes:

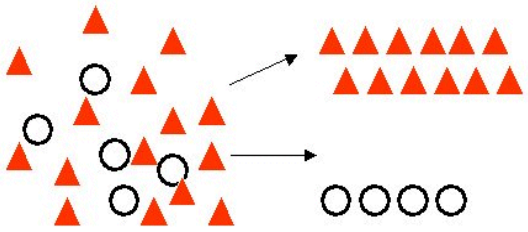
Datos	
Actualizar Tabla	Actualización de una tabla o vista.
Consulta SQL	Realización de una consulta SQL o PL/SQL a los datos de entrada al nodo, ofreciendo el resultado como salida.
Crear Tabla o Vista	Creación de una tabla o vista a partir de los datos de entrada del nodo.
Explorar Datos	Generación de estadísticas y gráficos a partir de los datos de entrada.
Gráfico	Generación de gráficos a partir de los datos de entrada.
Origen de Datos	Selección del origen de los datos, ya sea una tabla o una vista.
Transformaciones	
Agregado	Realización de agrupaciones (GROUP BY).

 Detalles de Filtrado de Columnas	Muestra los detalles del filtrado de columna.
 Filtrar Columnas	Selección de los atributos de la tabla/vista que queremos usar.
 Filtrar Filas	Selección de filas mediante el uso de alguna condición.
 Muestreo	Selección de una muestra de los datos de entrada a partir de diferentes criterios (aleatoriamente, desde-hasta…)
 Transform...	Transformación de los datos de entrada, ya sea cambio de tipo de dato o cambio de los valores de los datos.
 Unión	Unión de varias tablas o vistas (Join).
Modelos	
 Asociación	Creación de un modelo de asociación a partir de los datos de entrada.
 Clasificación	Creación de un modelo de clasificación a partir de los datos de entrada.
 Clusters	Creación de un modelo de clustering a partir de los datos de entrada.
 Detalles de Modelo	Visualizar los detalles de un modelo.
 Detección de Anomalías	Creación de un modelo para detección de anomalías a partir de los datos de entrada.
 Extracción de Funciones	Creación de un modelo para la extracción de funciones a partir de los datos de entrada.
 Modelo	Permite añadir modelos que no están en el flujo de trabajo actual, tales como modelos de otros flujos de trabajo o modelos creados mediante PL/SQL.
 Regresión	Creación de un modelo de regresión a partir de los datos de entrada.
Evaluación y aplicación	
 Aplicar	A partir de 2 entradas, un modelo y un conjunto de datos, este aplica el modelo a los datos y ofrece el resultado como salida del nodo.
 Probar	Realiza test a un modelo para probar su confianza.
Enlace de nodos	

Enlazar	Enlaza dos nodos entre sí.

3. Técnica de Clasificación

3.1. Descripción



La clasificación es uno de los tipos de problemas más importantes de la minería de datos. Para poder aplicarla, necesitamos un conjunto de datos caracterizado por algún atributo cuyo valor pertenece a diferentes clases. Por ejemplo, el atributo TIENE_COCHE está dividido en las clases SI y NO, o el atribulo COLOR puede estar dividido en las clases ROJO, VERDE y AZUL. La etiqueta de clase es un valor (simbólico) discreto y es conocido para cada objeto. El objetivo es construir los modelos de clasificación (a veces llamados clasificadores), que asignan la etiqueta de clase correcta a objetos que no tienen asignado ninguna.

Los modelos de clasificación son usados principalmente para el modelado predictivo. Por ejemplo, si tenemos una base de datos donde almacenamos las características visuales de los tumores encontrados en los distintos pacientes de un hospital y el resultado que se obtuvo después de analizarlo en el laboratorio (benigno o maligno) y teniendo en cuenta que el número de datos es suficiente y existe una correlación entre las características visuales de un tumor y su malignidad.

Con toda esta información se puede entrenar un modelo de clasificación, el cual posteriormente se podrá aplicar a los datos de los nuevos tumores encontrados en pacientes. El resultado final determinará en primer lugar, si se trata de un tumor de naturaleza benigna o maligna (clases), así como la probabilidad de que la predicción sea cierta. Esta información podría ayudar por ejemplo a priorizar a los pacientes en las lista de espera.

3.2. Ejemplo simple

Supongamos que una entidad bancaria tiene los siguientes datos sobre sus clientes:

ID	NAME	AGE	FOUNDS	CAR OWNERSHIP	HAS CHILDREN	HOUSE OWNERSHIP	MARITAL STATUS	CREDIT CARD
1	MARK	21	324	NO	NO	NO	SINGLE	**NO**
2	PAULA	45	13567	YES	YES	YES	MARRIED	**YES**
3	JIMMY	23	535	YES	NO	NO	SINGLE	

4	HELEN	67	15500	YES	YES	YES	WIDOWED	**YES**
5	PAUL	25	1500	YES	NO	NO	SINGLE	**NO**
6	JULIA	55	11345	YES	NO	NO	MARRIED	
7	SAM	37	18000	YES	NO	YES	SINGLE	**YES**
8	JON	53	17000	YES	YES	YES	MARRIED	**YES**
9	TOMAS	18	345	YES	NO	NO	SINGLE	

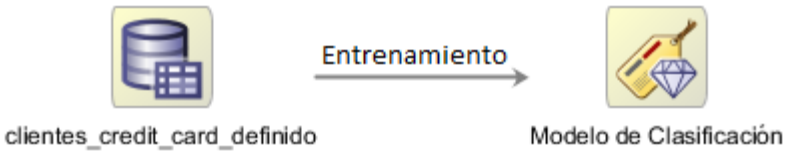
La entidad quiere saber, cuáles de sus clientes que no poseen tarjeta de crédito, aceptarían una. En la columna CREDIT_CARD se refleja si un cliente tiene tarjeta (YES) o si se le ha ofrecido y la ha rechazado (NO). Como vemos hay clientes a los cuales no se les ha ofrecido (JIMMY, JULIA, TOMAS).

En este caso, podemos usar la técnica de clasificación para predecir si un cliente aceptará o no la tarjeta de crédito.

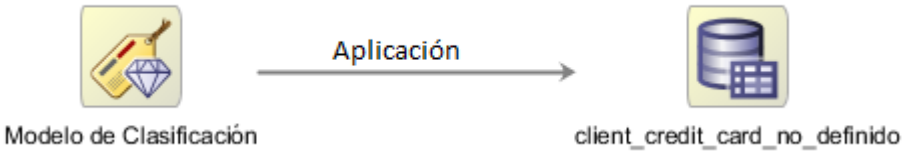
Para ello, seleccionamos los clientes que ya tienen ese atributo (clase) definido.

ID	NAME	AGE	FOUNDS	CAR OWNERSHIP	HAS CHILDREN	HOUSE OWNERSHIP	MARITAL STATUS	CREDIT CARD
1	MARK	21	324	NO	NO	NO	SINGLE	**NO**
2	PAULA	45	13567	YES	YES	YES	MARRIED	**YES**
4	HELEN	67	15500	YES	YES	YES	WIDOWED	**YES**
5	PAUL	25	1500	YES	NO	NO	SINGLE	**NO**
7	SAM	37	18000	YES	NO	YES	SINGLE	**YES**
8	JON	53	17000	YES	YES	YES	MARRIED	**YES**

Con estos datos seleccionados podemos entrenar un modelo de clasificación que posteriormente usaremos para clasificar clientes a los que no se les ha ofrecido la tarjeta.



Una vez entrenado el modelo, lo aplicamos a aquellos clientes que no tienen el atributo definido (JIMMY, JULIA, TOMAS).



Obteniendo la siguiente salida, donde se puede ver la predicción y la probabilidad de que se cumpla.

ID	NAME	CREDIT CARD PREDICTION	PROBABILITY
3	JIMMY	NO	**91%**
6	JULIA	YES	**82%**
9	TOMAS	NO	**85%**

3.3. Algoritmos de clasificación

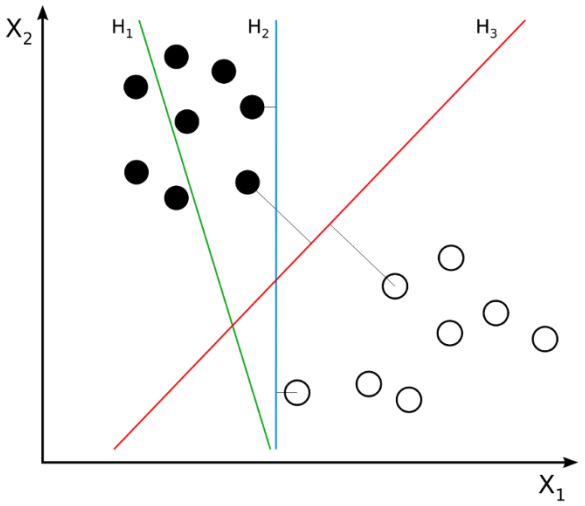
3.3.1. Support Vector Machine (SVM) – Maquina de Soporte Vectorial

3.3.1.1. Descripción

Dado un conjunto de puntos, subconjunto de un conjunto mayor (espacio), en el que cada uno de ellos pertenece a una de dos posibles categorías, un algoritmo basado en SVM construye un modelo capaz de predecir si un punto nuevo (cuya categoría desconocemos) pertenece a una categoría o a la otra.

La SVM busca un hiperplano, siendo una recta el caso más simple, que separe de forma óptima a los puntos de una clase de la de otra.

En el concepto de "separación óptima" es donde reside la característica fundamental de las SVM: este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia (margen) con los puntos que estén más cerca de él mismo. Por eso también a veces se les conoce a las SVM como clasificadores de margen máximo. De esta forma, los puntos del vector que son etiquetados con una categoría (clase) estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado.



Como se puede ver en la imagen, el hiperplano que separa a las 2 clases de puntos y que tiene mayor distancia media a ellos, es el H3 (rojo).

3.3.1.2. Opciones principales

Función de Núcleo: Selección del kernel Lineal o Gausiano.

3.3.2. Naive Bayes - Clasificador bayesiano ingenuo

3.3.2.1. Descripción

En teoría de la probabilidad y minería de datos, un clasificador Bayesiano ingenuo es un clasificador probabilístico fundamentado en el teorema de Bayes y algunas hipótesis simplificadoras adicionales.

En términos simples, un clasificador de Bayes ingenuo asume que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de cualquier otra característica, dada la clase variable. Por ejemplo, una fruta puede ser considerada como una manzana si es roja, redonda y de alrededor de 7 cm de diámetro. Un clasificador de Bayes ingenuo considera que cada una de estas características contribuye de manera independiente a la probabilidad de que esta fruta sea una manzana, independientemente de la presencia o ausencia de las otras características.

Para otros modelos de probabilidad, los clasificadores de Bayes ingenuo se pueden entrenar de manera muy eficiente en un entorno de aprendizaje supervisado. En muchas aplicaciones prácticas, la estimación de parámetros para los modelos Bayes ingenuo utiliza el método de máxima verosimilitud, en otras palabras, se puede trabajar con el modelo ingenuo de Bayes sin aceptar probabilidad bayesiana o cualquiera de los métodos bayesianos.

Una ventaja del clasificador de Bayes ingenuo es que solo se requiere una pequeña cantidad de datos de entrenamiento para estimar los parámetros (las medias y las varianzas de las variables) necesarias para la clasificación. Como las variables independientes se asumen, solo es necesario determinar las varianzas de las variables de cada clase y no toda la matriz de covarianza.

3.3.2.1. Opciones principales

Umbral de singleton: Especifica el umbral de un valor de atributo de un predictor dado. El número de instancias de un valor dado debe igualar o sobrepasar la fracción especificada o el valor se ignorará.

Umbral Pairwairs (por parejas): Especifica el umbral de una pareja de valores determinada de atributo y predictor. El número de instancias de una pareja de valores en particular debe igualar o sobrepasar la fracción especificada o la pareja se ignorará.

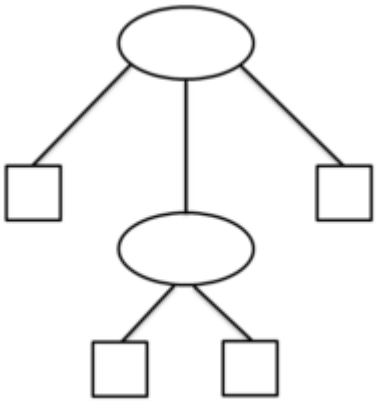
3.3.3. Decision Trees - Árbol de decisión

3.3.3.1. Descripción

Modelo de clasificación también conocido como ID3 que significa *"inducción mediante árboles de decisión"* que fue desarrollado por J. Ross Quinlan, capaz de tomar decisiones con gran precisión. Sistema de aprendizaje supervisado que aplica la estrategia *"divide y vencerás"* para hacer la clasificación, implementando métodos y técnicas para la realización de procesos inteligentes, representando así el conocimiento y el aprendizaje, con el propósito de automatizar tareas.

La salida del algoritmo ID3 se representa como un grafo en forma de árbol, cuyos componentes son los siguientes:

- Un nodo principal, llamado raíz, situado en la parte superior. De este nodo parten líneas hacia otros nodos inferiores, que a su vez, pueden hacer las veces de nodo raíz.
- Nodos terminales, como su nombre lo indica, son nodos donde termina el flujo y que ya no son raíz de ningún otro nodo. Estos nodos terminales deben contener una respuesta, o sea, la clasificación a la que pertenece el objeto que ha conducido hasta él.
- Los demás nodos representan preguntas con respecto al valor de uno de los atributos.



- Las líneas representan las posibles respuestas que los atributos pueden tomar.

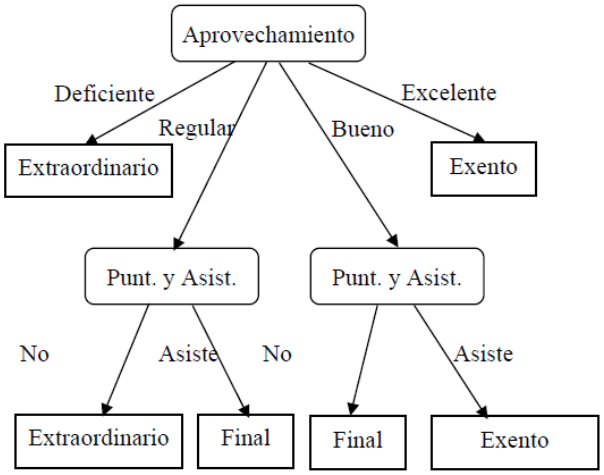
Cada objeto se representa con una lista de valores, el atributo y su valor, la cual constituye una descripción conjuntiva de ese objeto. El objeto debe ser etiquetado con la clase a la que pertenece.

La idea básica del ID3 es de determinar, para un conjunto de ejemplos dados, el atributo más importante, o sea, aquel que posea el mayor poder discriminatorio para dicho conjunto; éste atributo es usado para la clasificación de la lista de objetos, basados en los valores asociados con él mismo. Después de haber hecho la primera prueba de atributo, ésta, arrojará un resultado, el cual es en sí mismo un nuevo problema de aprendizaje de árbol de decisión, con la diferencia de que contará con menos ejemplos y un atributo menos, por lo que, cada atributo que se selecciona se descarta para la siguiente prueba.

Supongamos que tenemos una tabla con los siguientes datos.

Alumno	Punt. y asist.	Participación	Aprovechamiento	Nota
1	No asiste	Media	Excelente	Exento
2	Asiste	Alta	Bueno	Exento
3	No asiste	Media	Bueno	Final
4	No asiste	Baja	Bueno	Final
5	Asiste	Alta	Regular	Final
6	Asiste	Baja	Deficiente	Extraordinario
7	No asiste	Media	Regular	Extraordinario

Al aplicar el algoritmo, nos genera un árbol con la siguiente estructura. El cual ya podemos usar para predecir la nota de un alumno.



Por ejemplo, si tenemos un alumno con las siguientes características:

- Aprovechamiento: **Bueno**.
- Puntuación y Asistencia: **Asiste**.
- Participación: **Alta**.

Podríamos predecir que su nota es ***Exento***.

3.3.3.2. Opciones Principales

Profundidad máxima: Indica la profundidad máxima de árbol que se va a generar.

Porcentaje mínimo de registros en un nodo: Establece el porcentaje de número mínimo de registros por nodo.

Mínimo de registros en un nodo: Indica el número mínimo de registros devueltos.

Porcentaje mínimo de registros para una división: Establece el número mínimo de registros en un nodo padre, expresado como porcentaje del número total de registros empleados para entrenar el modelo. No se intenta dividir cuando el número de registros es inferior a este porcentaje.

Mínimo de registros para una división: Establece el número mínimo de registros en un nodo padre expresado como un valor. No se intenta dividir cuando el número de registros es inferior a este valor.

3.3.4. Generalized Linar Model (GLM) – Modelo Lineal Generalizado

3.3.4.1. Descripción

En estadística, el modelo lineal generalizado (MLG) es una generalización flexible de la regresión lineal ordinaria. Relaciona la distribución aleatoria de la variable dependiente en el experimento (la «función de distribución») con la parte sistemática (no aleatoria) (o «predictor lineal») a través de una función llamada la «función de enlace».

Los modelos lineales generalizados fueron formulados por John Nelder y Robert Wedderburn como una manera de unificar varios modelos estadísticos, incluyendo la regresión lineal, regresión logística y regresión de Poisson, bajo un solo marco teórico. Esto les permitió desarrollar un algoritmo general para la estimación de máxima verosimilitud en todos estos modelos. Esto puede ser naturalmente extendido a otros muchos.

3.3.4.2. Opciones principales

Nivel de confianza: El grado de verosimilitud, entre 0,0 y 1,0, del valor predicho para el objetivo en un intervalo de confianza calculado para el modelo.

Nombre de clase de referencia: Selección del atributo del cual queremos obtener las distintas clases.

Tratamiento de valores que faltan: En él se indica cómo se procesarán los valores nulos en los datos de entrada:

- **Modo de promedio:** sustituye los valores nulos de los atributos numéricos con el valor de la media y los valores de atributos categóricos con la moda.
- **Suprimir fila:** ignora las filas con valores nulos.

Regresión de resalto: Es una técnica que compensa la situación en la que existe un grado de correlación demasiado alto en las variables. Puede utilizar la opción **Auto** para permitir que el algoritmo controle el uso de esta técnica, o bien, puede controlarlo manualmente mediante las opciones **Desactivar** y **Activar**.

3.4. Ejemplo aplicado a Dataset de la Universidad de Irvine

En este ejemplo vamos a utilizar las técnicas de clasificación de la aplicación Oracle Data Miner para clasificar habitantes censados en EEUU en dos grupos. Siendo el primer grupo los que ganan más de 50.000$ anuales (>50K) y los del segundo grupo, aquellos que ganan menos de esa cantidad (<=50K). El dataset que ofrece la universidad de Irvine (http://archive.ics.uci.edu/ml/datasets/Census+Income) está compuesto por dos conjuntos de datos, el primero para entrenar el modelo de clasificación (el cual llamaremos CENSUS) y el segundo para aplicarle el modelo entrenado y predecir a que grupo pertenece cada usuario censado (CENSUS_APPLY).

Los atributos con los que vamos a trabajar son los siguientes:

ATRIBUTO	SIGNIFICADO	POSIBLES VALORES
Id	Identificador	Valor discreto
Age	Edad	Valor continuo
workclass	Tipo de trabajador	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked
Fnlwgt		Valor continuo
education	Educación	Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool
education-num		Valor continuo
marital-status	Estado matrimonial	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
occupation	Ocupación	Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces
relationship		Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried
Race	Raza	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
Sex	Género	Female, Male
capital-gain	Ingresos de fuentes de inversión	Valor continuo
capital-loss	Perdidas de fuentes de inversión	Valor continuo
hours-per-week	Número de horas de trabajo por semana	Valor continuo
native-country	País natal	United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands
income	Ingresos	>50K, <=50K

En la siguiente tabla dividida en dos niveles, tenemos una muestra del dataset CENSUS. Se puede apreciar que hay campos con valores indefinidos (?).

ID	AGE	WORK CLASS	FNLWGT	EDUCATION	EDUCATION _NUM	MARITAL _STATUS	OCCUPATION
1	30	Private	59496	Bachelors	13	Married-civ-spouse	Sales
2	32	?	293936	7th-8th	4	Married-spouse-absent	?
3	48	Private	149640	HS-grad	9	Married-civ-spouse	Transport-moving
4	42	Private	116632	Doctorate	16	Married-civ-spouse	Prof-specialty

ID	RELATIONSHIP	RACE	SEX	CAPITAL _GAIN	CAPITAL _LOSS	HOURS_PER _WEEK	NATIVE_ CONUNTRY	TARGET
1	Husband	White	Male	2407	0	40	United-States	<=50K
2	Not-in-family	White	Male	0	0	40	?	<=50K
3	Husband	White	Male	0	0	40	United-States	<=50K
4	Husband	White	Male	0	0	45	United-States	>50K

Para realizar una predicción más precisa, vamos a eliminar todas las filas que contengan algún atributo no definido. Para ello, después de agregar el dataset CENSUS al flujo de trabajo, vamos a ejecutar una consulta contra él, que elimine los datos anteriormente mencionados.

La consulta que ejecuta el nodo SQL contra el dataset CENSUS es la siguiente:

```
SELECT * FROM CENSUS_FINAL
WHERE
WORKCLASS NOT LIKE '%?' AND
EDUCATION  NOT LIKE '%?' AND
MARITAL_STATUS  NOT LIKE '%?' AND
OCCUPATION  NOT LIKE '%?' AND
RELATIONSHIP  NOT LIKE '%?' AND
RACE  NOT LIKE '%?' AND
SEX  NOT LIKE '%?' AND
NATIVE_COUNTRY  NOT LIKE '%?'
```

La salida del nodo SQL ya no contiene elementos con indeterminaciones. Por lo que podemos proceder a crear el modelo de clasificación.

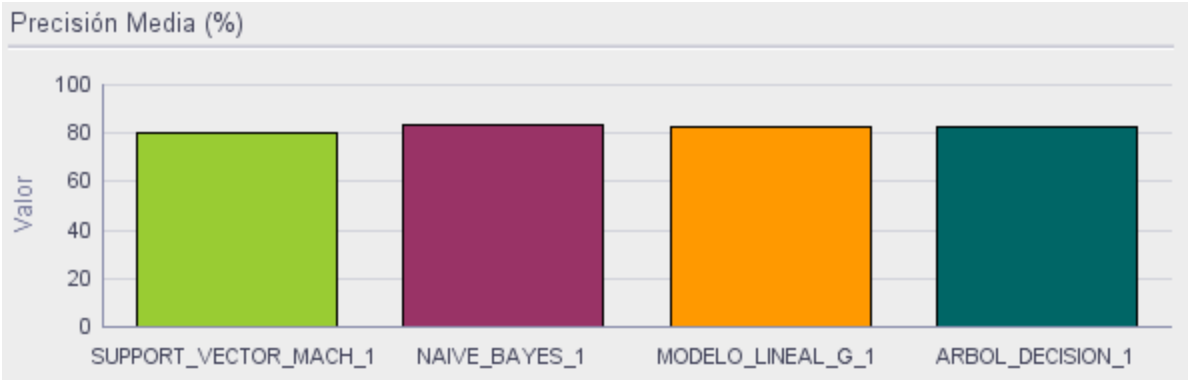
El modelo está configurado para realizar el entrenamiento con los cuatro algoritmos mencionados con anterioridad. De modo que, aunque en nuestro flujo de trabajo sólo aparezca un nodo de clasificación, realmente serían cuatro.

Una vez entrenado el modelo, evaluamos los resultados obtenidos por cada algoritmo. Para ello pulsamos con el botón secundario sobre el nodo “Modelo de clasificación” e indicamos que queremos comprobar los resultados de las pruebas. Estos resultados son los que genera el modelo de clasificación cuando se aplica a los datos con los que ha sido entrenado.

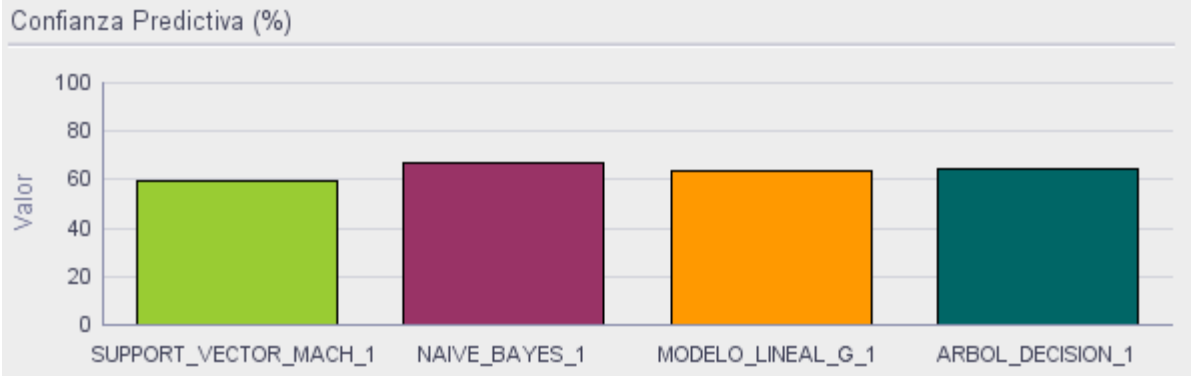
En nuestro caso, hemos indicado que se aplicaran al conjunto completo de datos. Sin embargo, el software nos permite seleccionar el conjunto de elementos que deseemos para el entrenamiento. Incluso también podríamos configurarlo para que por defecto realizara la prueba con un conjunto de datos distinto del que ha sido entrenado.

En la ventana se nos muestra la siguiente información:

La **precisión media** nos muestra el porcentaje de predicciones correctas.

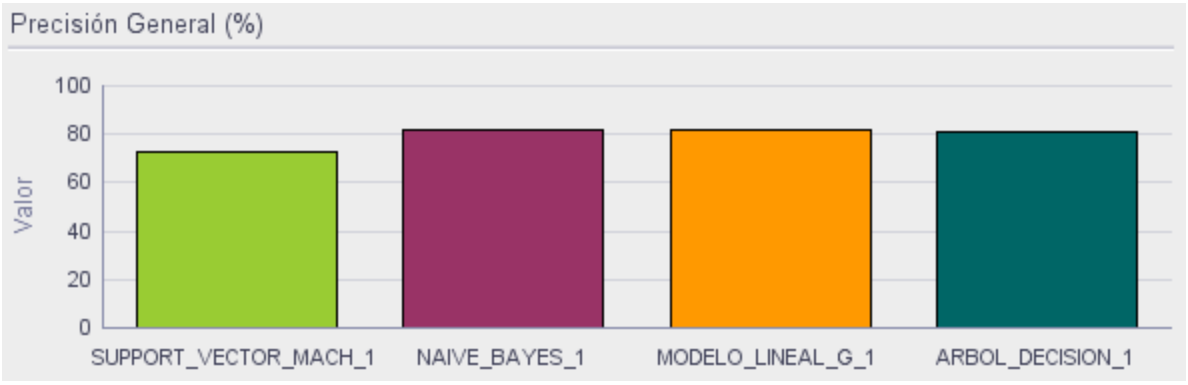


La **confianza predictiva** nos da una estimación global de la calidad del modelo entrenado. Indicando cuanto de mejor son las predicciones de cada algoritmo comparado con el modelo ingenuo.

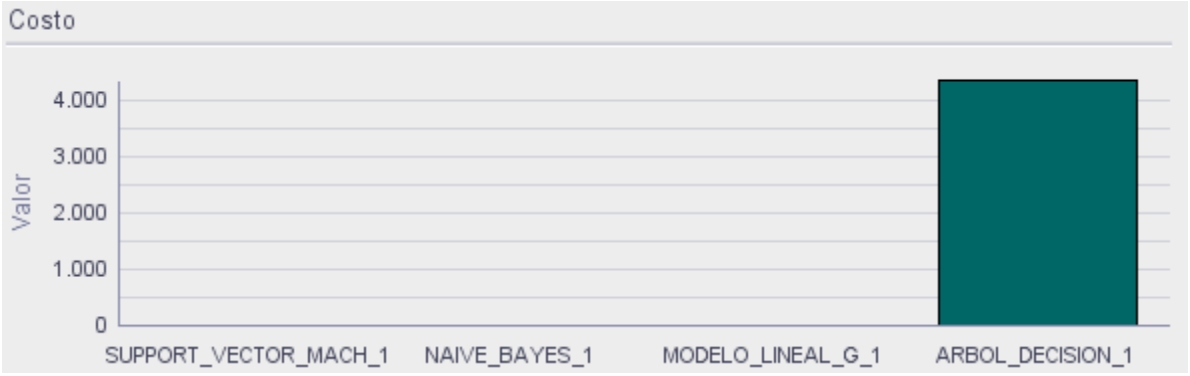


$$Confianza\ Predictiva = MAX\left[\left(1 - \frac{error\ del\ modelo}{error\ del\ modelo\ ingenuo}\right), 0\right] * 100$$

La **precisión general** es el porcentaje de la exactitud media por clase.



Aquí podemos observar el **coste** computacional generado por cada algoritmo.



Como podemos observar, los algoritmos ofrecen resultados de predicción muy equitativos. En cuanto al coste computacional, la generación del árbol de decisión se sobrepone a todos los demás de forma característica, debido a su estructura de árbol recursiva.

Para ver los resultados de cada algoritmo de forma más detallada, podemos hacer uso de la matriz de rendimiento, la cual se sitúa en la pestaña contigua a "Rendimiento":

Modelos	Porcentaje de Predicciones Correctas	Recuento de Predicciones Correctas	Total de Recuento de Casos
SUPPORT_VECTOR_MACH_1	72,308	8.622	11.924
NAIVE_BAYES_1	80,9963	9.658	11.924
MODELO_LINEAL_G_1	80,9292	9.650	11.924
ARBOL_DECISION_1	80,0822	9.549	11.924

El Modelo que implementa el algoritmo Naive Bayes se sobrepone mínimamente a los demás, mientras el algoritmo Support Vector Machine ofrece el peor resultado, 70% de aciertos, frente al 80% de los otros.

También podemos ver detalladamente los resultados obtenidos por cada algoritmo, por ejemplo los del algoritmo “Modelo Lineal General” son:

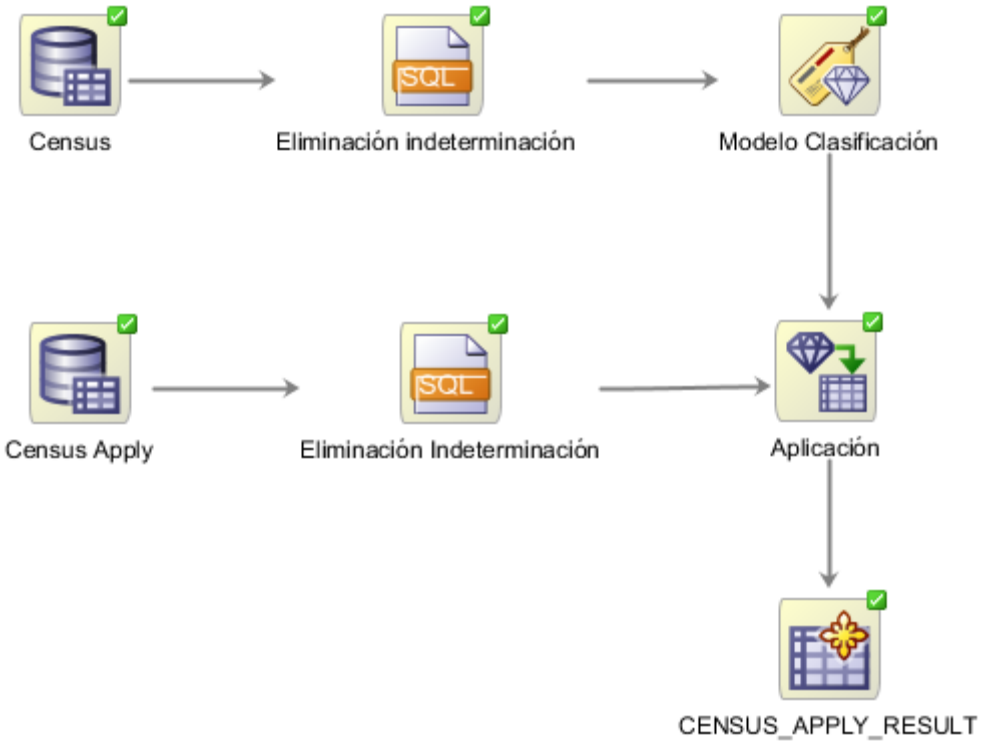
	<=50K	>50K
<=50K	17.970	4.683
>50K	1.150	6.358
Total	19.120	11.041
Porcentaje de Correctas	93,9854	57,5854
Costo		

Ahora procedemos a aplicar el modelo de clasificación al conjunto de datos CENSUS APPLY. Aunque este conjunto tenga definido el atributo INCOME (<=50K, >50K) por cada persona, debemos de imaginar que no lo tiene, ya que nuestro objetivo es predecirlo.

Este valor nos será útil al final para calcular la probabilidad de acierto de cada algoritmo. Por lo que añadimos al espacio de trabajo el dataset. Asimismo, eliminamos las filas de datos que contengan alguna indeterminación.

Una vez hecho esto añadimos el nodo “Aplicación” en el que vamos tener como entradas por un lado, el modelo de clasificación y por otro, los datos ya tratados del dataset CENSUS_APPLY.

Finalmente, la salida del nodo “aplicación”, será una tabla que contenga el identificador de la persona, el valor real del atributo INCOME, la predicción hecha por cada algoritmo y la probabilidad de que sea cierta. A continuación podemos ver cómo quedaría el espacio de trabajo.



Extracto de la tabla que genera la aplicación del modelo de clasificación:

ID_CENSUS	REAL VALUE	GLM_PRED	GLM_PROB	SVM_PRED	SVM_PROB	DT_PRED	DT_PROB	NV_PRED	NV_PROB
32.677	<=50K	<=50K	0,881	<=50K	0,7026	<=50K	0,8848	<=50K	0,99
32.678	<=50K	<=50K	0,9916	<=50K	0,9776	<=50K	0,9899	<=50K	1
32.680	<=50K	<=50K	0,7838	<=50K	0,5774	<=50K	0,9659	<=50K	0,9999
32.682	>50K	<=50K	0,6965	>50K	0,5844	<=50K	0,8046	<=50K	0,8086
32.683	<=50K	<=50K	0,9877	<=50K	0,9709	<=50K	0,9899	<=50K	1

Para calcular el porcentaje de aciertos de cada algoritmo se ha realizado el siguiente bloque PL/SQL:

```
DECLARE
  num_success_more NUMBER;
  num_success_minus NUMBER;
  num_total NUMBER;
  percent_success NUMBER;

BEGIN

-- NAIVE BAYES
  SELECT COUNT(*) INTO num_success_more FROM CENSUS_APPLY_RESULT WHERE
TARGET LIKE '%>50K' AND CLAS_NB_1_12_PRED LIKE '%>50K';
```

```
  SELECT COUNT(*) INTO num_success_minus FROM CENSUS_APPLY_RESULT WHERE
TARGET LIKE '%<=50K' AND CLAS_NB_1_12_PRED LIKE '%<=50K';
  SELECT COUNT(*) INTO num_total FROM CENSUS_APPLY_RESULT;

  percent_success := (num_success_more+num_success_minus)/num_total;

  DBMS_OUTPUT.put_line('Porcentaje de Aciertos Naive Bayes: ');
  DBMS_OUTPUT.put_line(percent_success);
  DBMS_OUTPUT.put_line('Porcentaje de Error Naive Bayes: ');
  DBMS_OUTPUT.put_line(1-percent_success);


-- SUPPORT VECTOR MACHINE
  SELECT COUNT(*) INTO num_success_more FROM CENSUS_APPLY_RESULT WHERE
TARGET LIKE '%>50K' AND CLAS_SVM_1_12_PRED LIKE '%>50K';
  SELECT COUNT(*) INTO num_success_minus FROM CENSUS_APPLY_RESULT WHERE
TARGET LIKE '%<=50K' AND CLAS_SVM_1_12_PRED LIKE '%<=50K';

  percent_success := (num_success_more+num_success_minus)/num_total;

  DBMS_OUTPUT.put_line('Porcentaje de Aciertos Support Vector Machine: ');
  DBMS_OUTPUT.put_line(percent_success);
  DBMS_OUTPUT.put_line('Porcentaje de Error Support Vector Machine: ');
  DBMS_OUTPUT.put_line(1-percent_success);


 -- MODELO LINEAL GENERAL
  SELECT COUNT(*) INTO num_success_more FROM CENSUS_APPLY_RESULT WHERE
TARGET LIKE '%>50K' AND CLAS_GLM_1_12_PRED LIKE '%>50K';
  SELECT COUNT(*) INTO num_success_minus FROM CENSUS_APPLY_RESULT WHERE
TARGET LIKE '%<=50K' AND CLAS_GLM_1_12_PRED LIKE '%<=50K';

  percent_success := (num_success_more+num_success_minus)/num_total;

  DBMS_OUTPUT.put_line('Porcentaje de Aciertos Modelo Lineal General: ');
  DBMS_OUTPUT.put_line(percent_success);
  DBMS_OUTPUT.put_line('Porcentaje de Error Modelo Lineal General: ');
  DBMS_OUTPUT.put_line(1-percent_success);




-- ARBOL DE DECISION
  SELECT COUNT(*) INTO num_success_more FROM CENSUS_APPLY_RESULT WHERE
TARGET LIKE '%>50K' AND CLAS_DT_1_12_PRED LIKE '%>50K';
  SELECT COUNT(*) INTO num_success_minus FROM CENSUS_APPLY_RESULT WHERE
TARGET LIKE '%<=50K' AND CLAS_DT_1_12_PRED LIKE '%<=50K';

  percent_success := (num_success_more+num_success_minus)/num_total;

  DBMS_OUTPUT.put_line('Porcentaje de Aciertos Arbol de Decision: ');
  DBMS_OUTPUT.put_line(percent_success);
  DBMS_OUTPUT.put_line('Porcentaje de Error Arbol de Decision: ');
  DBMS_OUTPUT.put_line(1-percent_success);

END;
```

Los resultados obtenidos son los siguientes:

ALGORITMO	ERROR
Naive Bayes	19%
Árbol de Decisión	20%
Support Vector Machine	31%
Modelo Lineal General	19%

Podemos ver, que los algoritmos que mejor han clasificado el conjunto de datos, son los mismos que tuvieron los mejores resultados en la fase de pruebas.

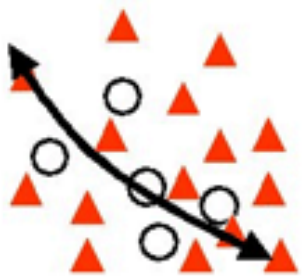
3.5. Ejemplo aplicado a Dataset de la Universidad de Irvine mediante PL/SQL

```
/* Creación de la tabla que contendrá las opciones del modelo */
CREATE TABLE census_class_settings
(setting_name VARCHAR2(30),setting_value VARCHAR2(4000));


/* Inserción de las opciones */
BEGIN
  /* Algoritmo: Árbol de decisión */
  INSERT INTO census_class_settings
  (setting_name, setting_value) VALUES
  (dbms_data_mining.algo_name, dbms_data_mining.algo_decision_tree);

  /* Preparación automática de los datos */
  INSERT INTO census_class_settings
  (setting_name, setting_value) VALUES
  (dbms_data_mining.prep_auto, dbms_data_mining.prep_auto_on);
END;
/



/* Creación del Modelo a partir de los datos situados en la
   tabla AUTO_MPG_PL */
BEGIN

  SYS.DBMS_DATA_MINING.CREATE_MODEL(
    model_name => 'CENSUS_CLASS_TREE_PL',
    mining_function => dbms_data_mining.classification,
    data_table_name => 'CENSUS',
    case_id_column_name => 'ID_CENSUS',
    target_column_name => 'INCOME',
    settings_table_name => 'census_class_settings'
  );

  END;
  /

/* Aplicación del modelo a los datos situados en la tabla
   CENSUS_APPLY y volcado del resultado en la tabla
   CENSUS_RESULT_PL */
BEGIN
```

```
  SYS.DBMS_DATA_MINING.APPLY(
    model_name => 'CENSUS_CLASS_TREE_PL',
    data_table_name => 'CENSUS_APPLY',
    case_id_column_name => 'ID_CENSUS',
    result_table_name => 'CENSUS_RESULT_PL'
  );
END;
/


/* Aplicación del modelo en tiempo real */
SELECT cust_id,
  PREDICTION(CENSUS_CLASS_TREE_PL USING *) Predicted_value,
  PREDICTION_PROBABILITY(CENSUS_CLASS_TREE_PL USING *) Probability
FROM mining_data_apply_v
WHERE rownum <= 10;
```

4. Técnica de Regresión

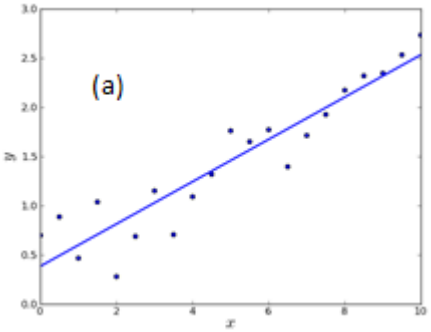
4.1. Descripción

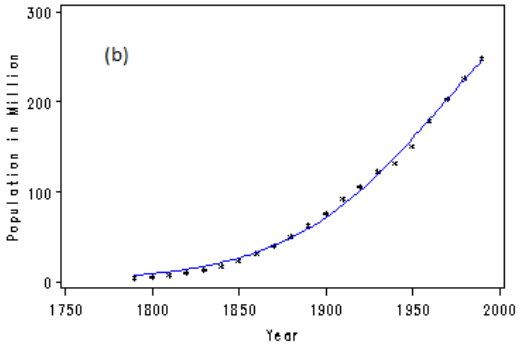


La regresión es una técnica de minería de datos utilizada para predecir valores continuos. Tradicionalmente es usada para crear una relación entre un atributo predictor y un atributo objetivo. Son muchas las similitudes entre ambas técnicas. En ambas se va a determinar un valor objetivo a partir de múltiples atributos. Además, como veremos a continuación los dos algoritmos usados para la creación de regresiones, también son usados en las técnicas de clasificación.

Fundamentalmente existen dos tipos de regresión: la regresión lineal (a) y la regresión no lineal (b).

Un ejemplo de regresión podría ser el siguiente: tenemos una tabla donde se indica el tanto por ciento del PIB que representa las exportaciones en España, desde el año 1980 hasta la actualidad. Con estos datos podríamos crear un modelo de regresión que nos permitiera realizar predicciones para los próximos años.



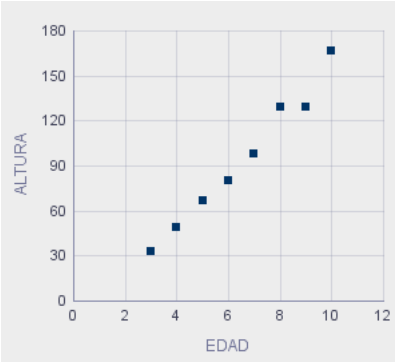


4.2. Ejemplo simple

Imaginemos lo siguiente: hace algún tiempo, se plantó un ejemplar de pino de la especie *Pinus Durangensis*. Los datos observados en la siguiente tabla, muestran la altura del ejemplar desde los tres años de edad hasta este momento:

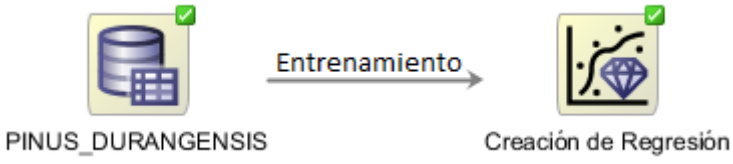
PINUS DURANGENSIS	
Edad (años)	**Altura (cm)**
3	33,5
4	49,38
5	66,6
6	80,22
7	98
8	129,33
9	129,44
10	166,33

(Fuente:http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0186-32312009000100008)



(Como podemos observar en la gráfica entre los atributos edad y altura existe una correlación lineal muy clara)

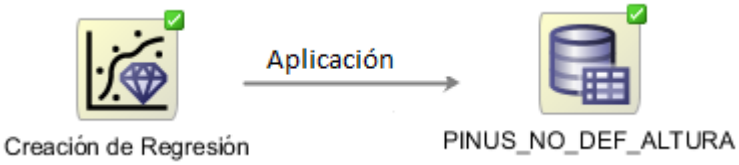
Ahora queremos predecir cuál será la altura de nuestro pino los próximos cinco años. Para ello vamos a crear un modelo de regresión con los datos que se han recopilado.



Una vez creado el modelo, necesitamos una tabla que contenga los cinco próximos años y la altura no definida, tal que así:

PINUS DURANGENSIS (PRED)	
Edad (años)	**Altura (cm)**
11	?
12	?
13	?
14	?
15	?

A esta tabla se le aplicará el modelo de regresión anterior.



Al aplicar el modelo obtenemos la predicción de cada atributo.

PINUS DURANGENSIS (PRED)	
Edad (años)	**Altura (cm)**
11	176,39
12	194,6767
13	212,9633
14	231,25
15	249,5367

4.3. Algoritmos de Regresión

4.3.1. Generalized Linear Model (GLM) – Modelo Lineal Generalizado

(Mismo algoritmo modelos de clasificación)

4.3.2. Support Vector Machine (SVM) – Maquina de Soporte Vectorial

(Mismo algoritmo modelos de clasificación)

4.4. Ejemplo aplicado a Dataset de la Universidad de Irvine

Para esta técnica se ha elegido un dataset donde se relaciona un conjunto de características de un vehículo con su consumo (https://archive.ics.uci.edu/ml/datasets/Auto+MPG). Es decir, vamos a predecir el consumo medio de un vehículo a partir de datos como el número de caballos, su peso o el número de cilindros.

Nuestro valor objetivo será el atributo **mpg** (miles per gallon).

El dataset ha sido modificado para introducirle un ID a cada elemento, ya que aunque en él se indica que el campo **car_name** es único para cada registro, esto no es cierto. Esto es debido a que aparecen distintos registros con el mismo nombre del modelo pero producidos en distinto año. Además se han eliminado 6 registros que contenían el atributo **horsepower** indefinido.

Finalmente se han extraído 10 tuplas aleatoriamente, que posteriormente se usarán para aplicarles el modelo de regresión que generemos.

En la siguiente tabla se muestra el conjunto de atributos con su significado y tipo de dato, que usaremos para construir nuestro modelo de regresión:

ATRIBUTO	SIGNIFICADO	POSIBLES VALORES
Id	Identificador	Valor continuo
mpg	Millas por galón	Valor discreto
cylinders	Número de cilindros	Valor discreto
displacement	Cilindrada	Valor continuo
horsepower	Número de caballos	Valor continuo
weight	Peso del vehículo	Valor continuo
acceleration	Aceleración	Valor continuo
model year	Año del modelo	Valor discreto
Origin	Origen	Valor discreto
car name	Nombre del coche	Cadena de texto

En la siguiente tabla dividida en dos niveles podemos ver un ejemplo de la información que contiene el dataset:

ID	MPG	CYLINDERS	DISPLACEMENT	HORSEPOWER	WEIGHT
1	18	8	307	130	3504
2	15	8	350	165	3693
3	18	8	318	150	3436

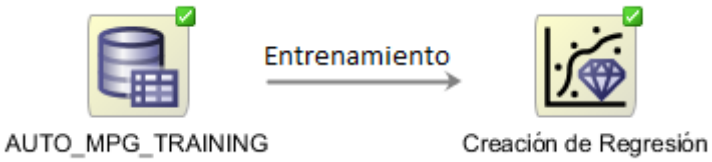
4	16	8	304	150	3433
5	17	8	302	140	3449
6	15	8	429	198	4341

ID	ACCELERATION	MODEL YEAR	ORIGIN	CAR NAME
1	12	1970	1	chevrolet chevelle malibu
2	11,5	1970	1	buick skylark 320
3	11	1970	1	plymouth satellite
4	12	1970	1	amc rebel sst
5	10,5	1970	1	ford torino
6	10	1970	1	ford galaxie 500

Para crear nuestro modelo de regresión, primero vamos a crear una tabla llamada AUTO_MPG_TRAINIG. Dicha tabla contendrá la información del dataset a excepción de los diez registros comentados anteriormente.

El siguiente paso es añadir a nuestro proyecto de minería de datos el nodo “Origen de Datos” donde seleccionaremos como origen nuestra tabla AUTO_MPG_TRAINIG y el nodo “Regresión”. Unimos los 2 nodos y pulsamos sobre el nodo “Regresión” donde queremos usar todos los campos para crear el modelo, a excepción del campo car_name, el cual poca información nos aporta sobre el consumo del coche. De igual forma vamos a indicar en las propiedades del nodo de regresión, que en el momento de realizar las pruebas a nuestro modelo, tome todo el conjunto de datos, ya que por defecto Oracle selecciona un conjunto compuesto por el 40% de la muestra dada para la creación del modelo.

Como en este caso el número de registros es de solo 382 elementos, realizar las pruebas sobre todo el conjunto nos ofrecerá una mejor visión del modelo.

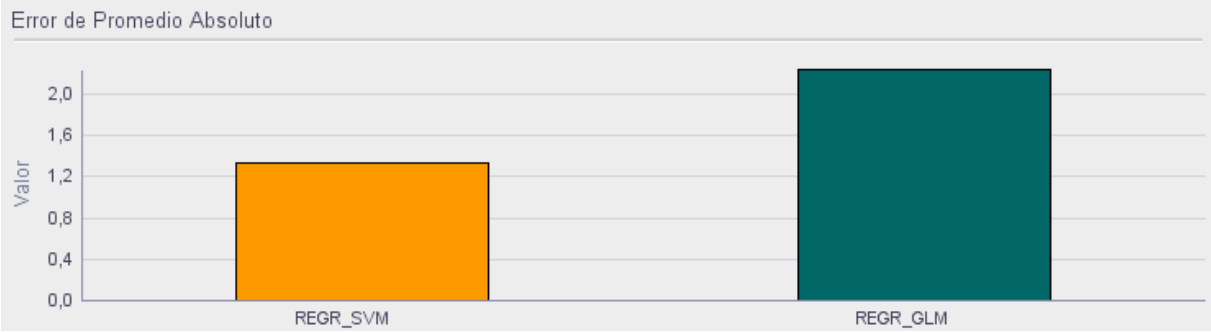


Una vez realizado el entrenamiento, procedemos a comparar los resultados ofrecidos por los 2 algoritmos de regresión:

La **confianza predictiva** nos da una estimación global de la calidad del modelo entrenado. Indicando cuanto de mejor son las predicciones de cada algoritmo comparado con el modelo ingenuo.

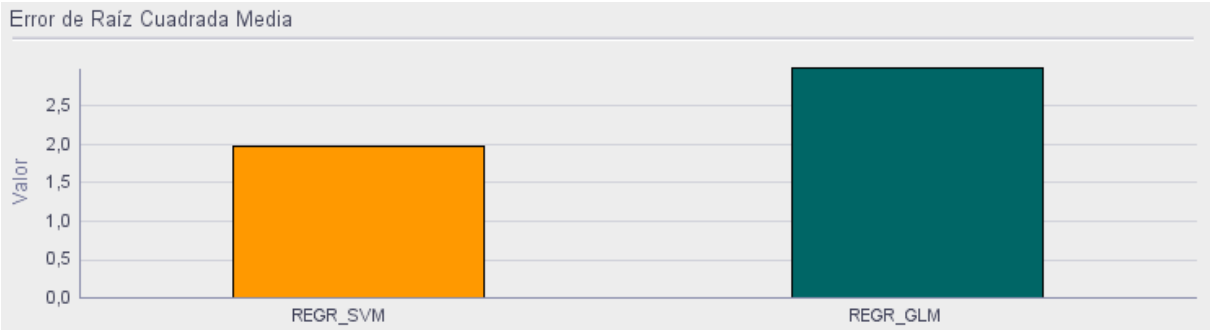
El **error de promedio absoluto** es la media de los valores absolutos de los valores residuales (diferencia entre el valor real de la medida y el valor predicho).

$$\frac{1}{n} * \sum_{1}^{n} |r - p|$$



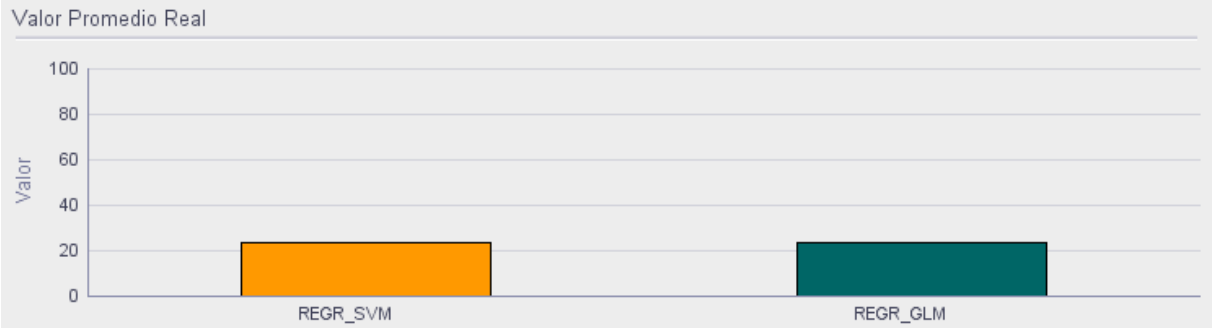
El **error de raíz cuadrada media** es la raíz cuadrada de la media de los valores residuales al cuadrado.

$$\sqrt{\frac{1}{n} * \sum_{1}^{n} (r - p)^2}$$



El **valor promedio previsto** indica la media de los valores que han sido predichos. En este caso, vemos que los dos algoritmos ofrecen una media prácticamente similar: 23,38 y 23,32 MPG (Milles per gallon) respectivamente.

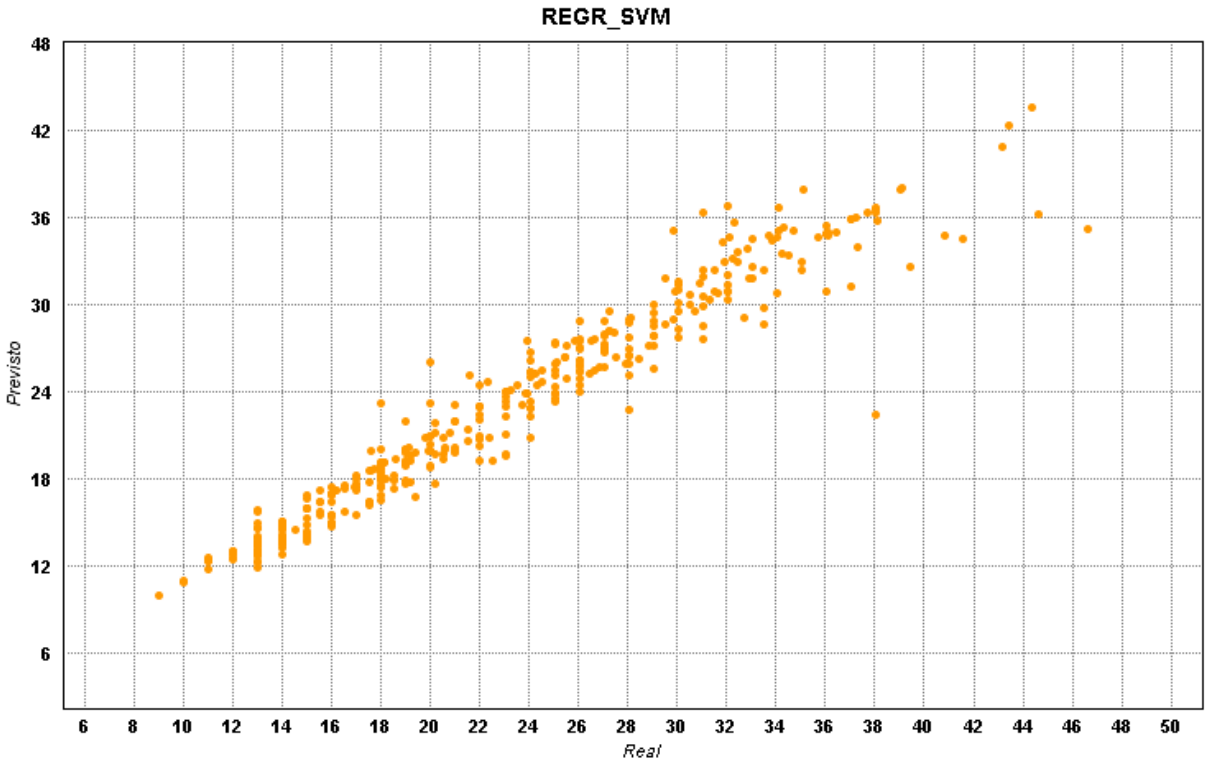
El **valor promedio real** nos muestra la media de los valores reales. Aunque aquí observamos una medida por cada algoritmo, las dos son exactamentes las mismas (23,38 MPG) ya que la aplicación de un algoritmo u otro no influye en la media de los valores reales.

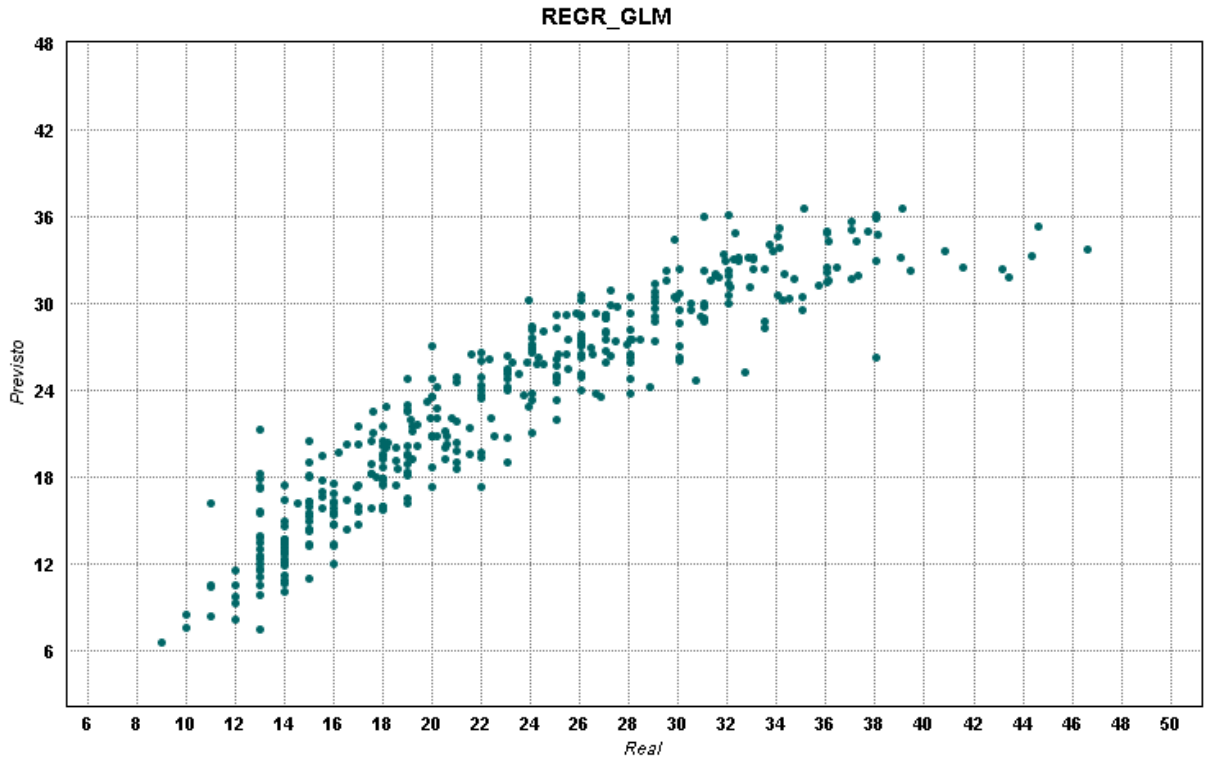


A partir de los datos de las dos últimas gráficas, podemos deducir que el funcionamiento de los dos algoritmos es bastante bueno. Los dos muestran una media de valores predichos similar a la media de los valores reales.

El hecho de que las medias de las predicciones de los algoritmos coincidan con la media de los valores reales, no puede ser tomado únicamente como método para la selección de un algoritmo u otro, ya que las predicciones superiores al valor real se pueden compensar con las predicciones inferiores al valor real.

También podemos ver cuál es el mejor modelo, generando las siguientes gráficas en Oracle Data Miner. Las gráficas representan en el eje X el consumo medio real de un vehículo (MPG) frente al eje Y, donde se representa el valor que ha predicho el algoritmo para ese mismo vehículo.





Si el funcionamiento de los algoritmos fuera ideal, los puntos deberían situarse sobre la bisectriz del cuadrante, es decir, el valor real y el valor previsto deberían de ser los

mismos (X = Y). Por lo que para comparar los dos algoritmos solo debemos fijarnos en cuál de las dos nubes de puntos se sitúa más sobre la bisectriz.

Al igual que la conclusión obtenida del análisis de las 5 gráficas iniciales, podemos observar que el algoritmo Support Vector Machine (SVM) tiene los puntos ligeramente más cercanos a la bisectriz, que el algoritmo Generalized Linar Model (GLM).

Una vez generado nuestro modelo, procedemos a aplicarlos a los datos que extrajimos con anterioridad. Como hemos visto, el algoritmo Support Vector Machine (SVM) es el que ofrece mejores predicciones, así que en un caso aplicado a la vida real sería el más adecuado. Sin embargo, nosotros vamos a utilizar los dos, para poder comparar los resultados que ofrecen.

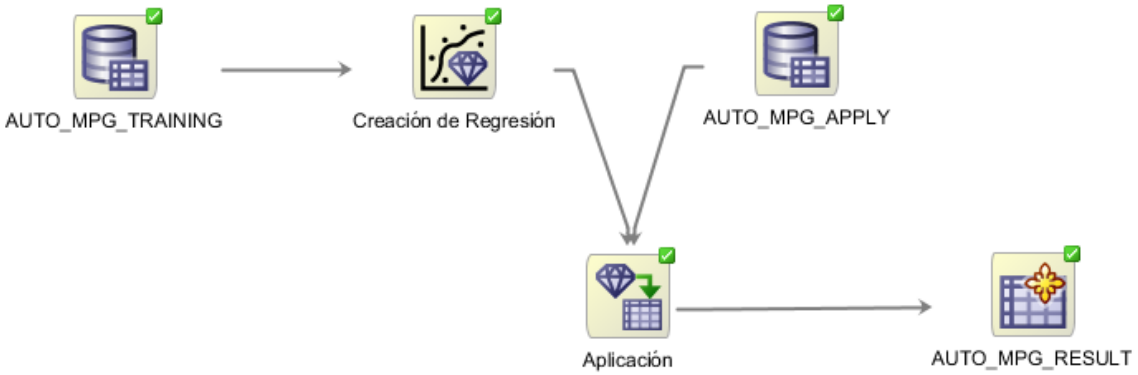
A continuación, se muestra la tabla que contiene los modelos de coches de los cuales pretendemos predecir su consumo medio. Hemos de recordar que aunque en la tabla aparece el campo **MPG** definido, debemos de imaginar que carecemos de dicho dato.

ID	MPG	CYLINDERS	DISPLACEMENT	HORSEPOWER	WEIGHT
5	17	8	302	140	3.449
15	24	4	113	95	2.372
69	12	8	350	160	4.456
376	36	4	107	75	2.205
389	44	4	97	52	2.130
391	28	4	120	79	2.625
348	34,4	4	98	65	2.045
273	20,3	5	131	103	2.830
269	21,1	4	134	95	2.515
241	21,5	4	121	110	2.600

ID	ACCELERATION	MODEL YEAR	ORIGIN	CAR NAME
5	10,5	1.970	1	ford torino
15	15	1.970	3	toyota corona mark ii
69	13,5	1.972	1	oldsmobile delta 88 royale
376	14,5	1.982	3	honda accord
389	24,6	1.982	2	vw pickup
391	18,6	1.982	1	ford ranger
348	16,2	1.981	1	ford escort 4w
273	15,9	1.978	2	audi 5000
269	14,8	1.978	3	toyota celica gt liftback
241	12,8	1.977	2	bmw 320i

Añadimos el dataset AUTO_MPG_APPLY que contiene la información de la tabla anterior a nuestro flujo de trabajo, junto con el nodo Aplicación. Indicamos al nodo aplicación cual es el modelo a aplicar y cuáles son los datos a los que será aplicado

el modelo (AUTO_MPG_APPLY). En las propiedades del nodo aplicación, modificamos la salida, incluyendo además del ID de cada modelo de coche, las predicciones, el atributo CAR_NAME y el atributo MPG. Gracias a eso podemos comparar el valor real con el valor predicho por los algoritmos. La salida la registraremos en una nueva tabla llamada AUTO_MPG_RESULT.



A continuación podemos observar la salida, una vez ha sido aplicado el modelo de regresión.

ID	CAR NAME	MPG	SVM_PRED	GLM_PRED
5	ford torino	17	17,8259	15,8177
15	toyota corona mark ii	24	25,1418	24,6598
69	oldsmobile delta 88 royale	12	12,8834	11,6781
376	honda accord	36	35,3011	34,8871
389	vw pickup	44	40,0978	34,3963
391	ford ranger	28	29,5363	29,7633
348	ford escort 4w	34,4	36,3737	32,4613
273	audi 5000	20,3	29,6276	30,8909
269	toyota celica gt liftback	21,1	26,4354	30,0268
241	bmw 320i	21,5	27,3814	27,1457

Con la siguiente consulta SQL podemos determinar el error cometido en cada predicción por cada uno de los algoritmos:

```
SELECT id, car_name, ABS(mpg - glm_pred) error_glm, ABS(mpg - svm_pred)
error_svm
FROM auto_mpg_result;
```

ID	CAR NAME	GLM_ERROR	SVM_ERROR
5	ford torino	1,18225798	0,82585114
15	toyota corona mark ii	0,6597903	1,14176734
69	oldsmobile delta 88 royale	0,32187316	0,8833714

376	honda accord	1,11289128	0,69888697
389	vw pickup	9,60370002	3,90224097
391	ford ranger	1,76334697	1,53634369
348	ford escort 4w	1,93865802	1,9736886
273	audi 5000	10,5908585	9,32758151
269	toyota celica gt liftback	8,92676427	5,33536647
241	bmw 320i	5,64570742	5,88141659

También podemos calcular el error promedio absoluto cometido por cada algoritmo:

```
SELECT sum(ABS(mpg - glm_pred))/count(id) ERROR_MEDIO_GLM,
sum(ABS(mpg - svm_pred))/count(id) ERROR_MEDIO_SVM
FROM auto_mpg_result;
```

GLM_ERROR_PROMEDIO_ABSOLUTO	SVM_ERROR_ PROMEDIO_ABSOLUTO
4,17458479%	3,15065147%

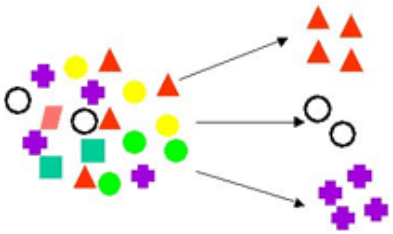
4.5. Ejemplo aplicado a Dataset de la Universidad de Irvine mediante PL/SQL

```
/* Creación de una vista con los atributos que vamos a usar para
   la creación del modelo de regresión */
CREATE or REPLACE VIEW AUTO_MPG_PL AS SELECT
    ID,
    MPG,
    CYLINDERS,
    DISPLACEMENT,
    HORSEPOWER,
    WEIGHT,
    ACCELERATION,
    MODEL_YEAR,
    ORIGIN
FROM auto_mpg_training;


/* Creación de la tabla que contendrá las opciones del modelo */
CREATE TABLE AUTO_MPG_REG_SET (
    SETTING_NAME VARCHAR(30),
    SETTING_VALUE VARCHAR(4000)
  );


/* Inserscion de las opciones */
BEGIN

  /* Algoritmo: Support Vector machine */
  INSERT INTO AUTO_MPG_REG_SET (SETTING_NAME, SETTING_VALUE) VALUES
  (SYS.dbms_data_mining.ALGO_NAME,
SYS.dbms_data_mining.ALGO_SUPPORT_VECTOR_MACHINES);

  /* Modo: Lineal */
```

```
  INSERT INTO AUTO_MPG_REG_SET (SETTING_NAME, SETTING_VALUE) VALUES
  (SYS.dbms_data_mining.SVMS_KERNEL_FUNCTION, SYS.dbms_data_mining.SVMS_LINEAR);


  /* Preparación automatica de los datos */
  INSERT INTO AUTO_MPG_REG_SET (SETTING_NAME, SETTING_VALUE) VALUES
  (SYS.dbms_data_mining.PREP_AUTO, SYS.dbms_data_mining.PREP_AUTO_ON);

END;
/

/* Creación del Modelo a partir de los datos situados en la
   vista AUTO_MPG_PL */
BEGIN
  SYS.dbms_data_mining.CREATE_MODEL(
    MODEL_NAME => 'AUTO_MPG_REGRESSION',
    MINING_FUNCTION => DBMS_DATA_MINING.REGRESSION,
    DATA_TABLE_NAME => 'AUTO_MPG_PL',
    CASE_ID_COLUMN_NAME => 'ID',
    TARGET_COLUMN_NAME => 'MPG',
    SETTINGS_TABLE_NAME => 'AUTO_MPG_REG_SET'
  );
END;
/


/* Aplicación del modelo a los datos situados en la tabla
   AUTO_MPG_TRAINING y volcado del resultado en la tabla
   AUTO_MPG_RESULT_PL */
BEGIN
  SYS.dbms_data_mining.APPLY(
    MODEL_NAME => 'AUTO_MPG_REGRESSION',
    DATA_TABLE_NAME => 'AUTO_MPG_TRAINING',
    CASE_ID_COLUMN_NAME => 'ID',
    RESULT_TABLE_NAME => 'AUTO_MPG_RESULT_PL'
  );
END;
/
```

5. Técnica de Clustering

5.1. Descripción

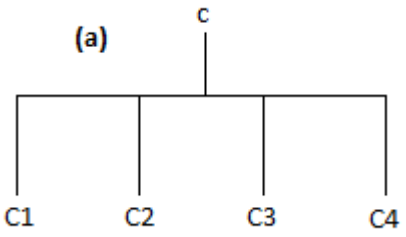


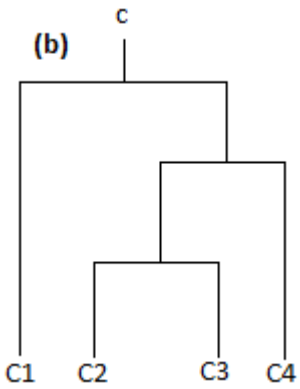
El clustering es una técnica de minería de datos que consiste en la división de un conjunto de elementos en conjuntos más pequeños llamados clusters, los cuales contienen datos que están relacionados entre sí de alguna forma.

Los algoritmos de clustering que vamos a utilizar nos permiten seleccionar el número de clusters que queremos obtener de los datos que disponemos. Cuando los algoritmos nos ofrecen los resultados obtenidos, no nos revelan ninguna información sobre por qué están organizados así. Solo nos muestran un conjunto de reglas que determinan cuando un elemento pertenece un clúster u otro.

Así que para utilizar la técnica de clustering podemos usar dos puntos de vista; el primero sería observar el contenido de los clúster e intentar identificar qué representa cada uno (caso que se verá en el próximo ejemplo); el segundo método consistiría en seleccionar un elemento cuyas características nos interesen y al realizar el clustering observar en que clúster está ubicado. De esta forma, sabremos que los elementos que lo acompañan tienen características similares.

Los clusters pueden están organizados en un único nivel (a) o jerárquicamente (b). Los algoritmos que vamos a utilizar en Oracle Data Miner son del segundo tipo. Lo que nos permite obtener algo más de información acerca de las relaciones entre los clusters, como por ejemplo saber si un clúster es un subconjunto de otro.





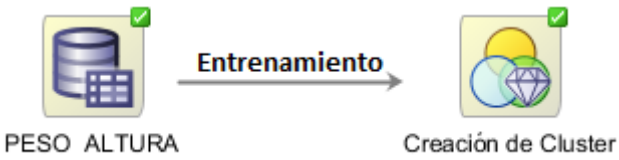
5.2. Ejemplo simple

Con el siguiente ejemplo comprenderemos fácilmente cómo funciona la técnica de clustering y cómo podríamos interpretar sus resultados. El ejemplo es muy simple y fijándonos brevemente en los datos, nosotros mismos podríamos identificar cuáles son los clusters en los que se podría dividir el conjunto de datos. Sin embargo, debemos de tener en cuenta que esta técnica es utilizada usualmente en conjunto de datos muy extensos en los que a simple vista no se ve relación alguna entre unos y otros.

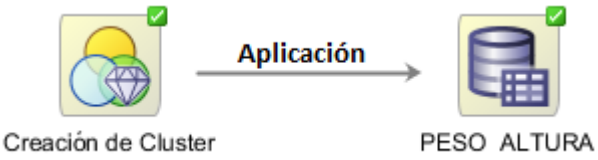
Partimos de una tabla donde se indican los pesos y alturas de nueve individuos:

ID	PESO	ALTURA
9	96,46	2,08
1	10,2	0,76
4	50,99	1,54
2	12,9	0,88
6	62,94	1,68
7	85,67	1,96
3	15,1	0,96
8	80	2
5	51,2	1,6

Entrenamos un modelo de clustering a partir de estos datos. En este caso suponemos que le hemos indicado al algoritmo que deseamos dividir el conjunto en tres clusters.



Una vez entrenado el modelo, podemos aplicarlo a nuevos datos o a los mismos datos con lo que lo hemos entrenado. Con la finalidad de que nos indique a que clúster pertenece cada elemento de la tabla.

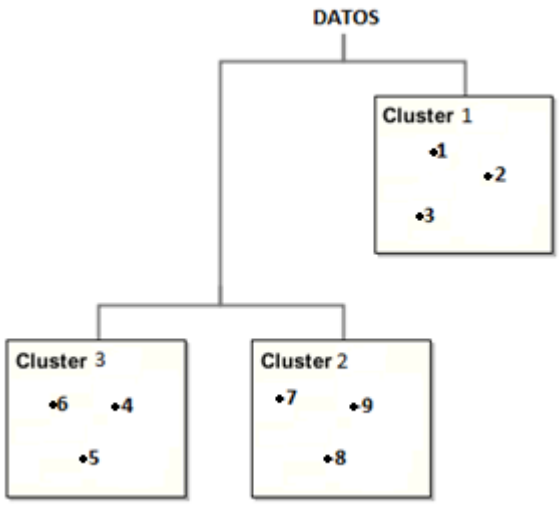


Como resultado obtenemos la siguiente tabla, en la que hemos ordenado los elementos agrupándolos por clúster.

ID	ALTURA	PESO	CLUTER
1	0,76	10,2	1
2	0,88	12,9	1
3	0,96	15,1	1
4	1,54	50,99	3
5	1,6	51,2	3
6	1,68	62,94	3
7	1,96	85,67	2
8	2	80	2
9	2,08	96,46	2

Si observamos los datos que contiene cada clúster, podemos percatarnos de qué individuos están representados en cada conjunto. Por ejemplo, en el clúster número uno podemos ver que los pesos y alturas pertenecen a niños de muy temprana edad, el número dos a adolescentes o adultos de mediana estatura y el número tres podemos ver representados a adultos de estatura elevada.

Además, como se mencionó anteriormente, los algoritmos que utiliza Oracle ofrecen una estructura de clusters jerárquica. Con la que obtenemos un esquema con las mismas características que el siguiente:



En el interior de cada clúster se encuentran los identificadores de los individuos de la tabla anterior. A partir de esta estructura podemos obtener algo de más información de la que podemos extraer de la tabla. Como que los clusters 2 y 3 tienen más relación entre sí (adultos), que con el clúster 1 (niños de edad temprana).

5.3. Algoritmos de Clustering

5.3.1. K-Means

5.3.1.1. Descripción

El algoritmo K-means de Oracle identifica los clústeres que se producen de forma natural en una población de datos. K-means es un algoritmo de agrupación en clústeres basado en la distancia que divide los datos en un número de clústeres predeterminado (siempre que haya suficientes casos distintos). Los algoritmos basados en la distancia confían en una métrica de distancia (función) para calcular la similitud entre los puntos de datos.

Los puntos de datos se asignan al clúster más próximo en función de la métrica de distancia empleada.

El algoritmo K-means admite clústeres jerárquicos, trata atributos numéricos y categóricos, y divide la población en el número de clústeres especificado por el usuario.

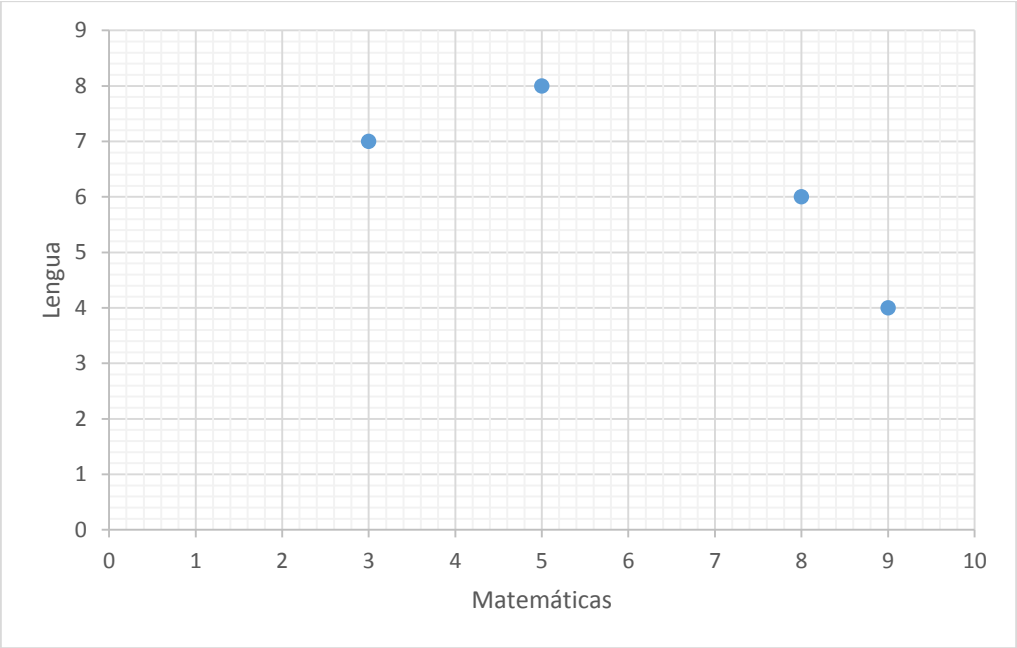
Oracle Data Miner proporciona información detallada, reglas y valores centroides del clúster, y se puede utilizar para puntuar una población en relación con su pertenencia a un clúster.

Como el funcionamiento del algoritmo en un caso básico es bastante simple, vamos a proceder a mostrar un ejemplo completo de su ejecución.

Aquí podemos ver el diagrama de flujo del algoritmo:

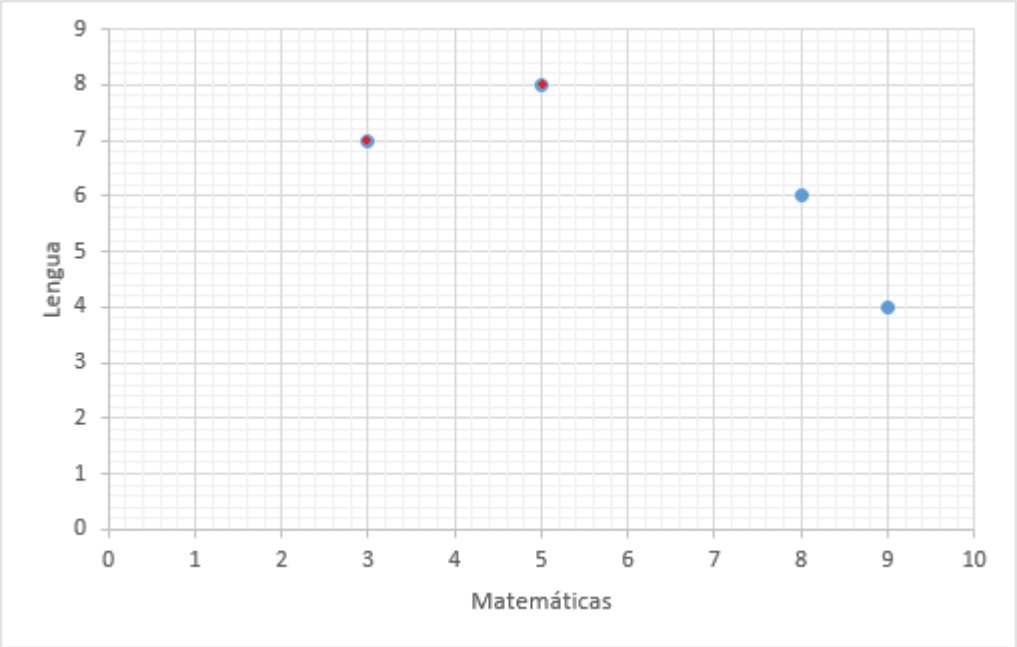
En la siguiente tabla tenemos a cuatro alumnos con sus distintas notas en matemáticas y lengua. Vamos a clasificarlos en dos clusters.

	Matemáticas	Lengua
Juan	9	4
Luis	3	7
José	8	6
Ángela	5	8



Iteración 1:

Seleccionamos dos centroides aleatoriamente, en este caso hemos elegido a c1 (3,7) y c2 (5,8).



Calculamos la distancia de cada punto a cada uno de los centroides utilizando la distancia euclidea.

$$d_E(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

	Juan (9,4)	Luis (3,7)	José (8,6)	Ángela (5,8)
Centroide 1 (3,7)	$3\sqrt{5}$	**0**	$\sqrt{26}$	$\sqrt{5}$
Centroide 2 (5,8)	$\mathbf{4\sqrt{2}}$	$\sqrt{5}$	$\mathbf{\sqrt{13}}$	**0**

Seleccionamos para cada individuo el centroide que le es más cercano (en negrita).

En este punto los clusters serían los siguientes:

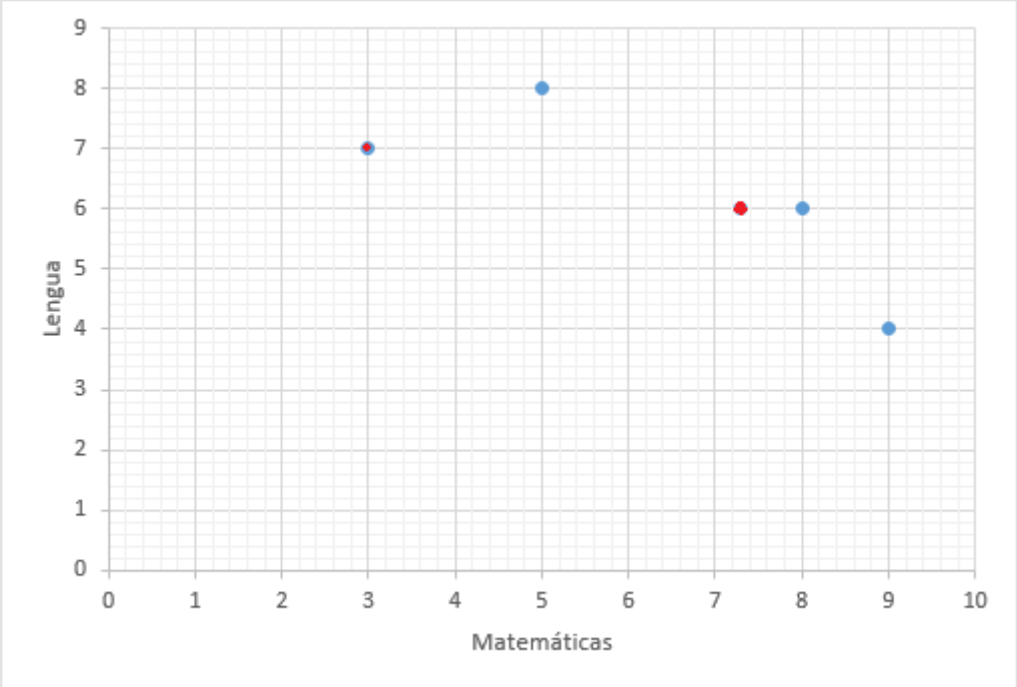
Cluster1 = {Luis}
Cluster2 = {Juan, José, Ángela}

Iteración 2:

Recalculamos los centroides, para ello obtenemos el punto medio a partir de los puntos que forman al clúster.

Centroide1: (3,7) ya que solo existe ese punto en el clúster.

Centroide2: ((9+8+5)/3, (4+6+8)/3) = (22/3,6)



Volvemos a calcular la distancia entre los puntos y los centroides, con la nueva ubicación de los centroides y seleccionamos el centroide más cercano en cada punto:

	Juan (9,4)	Luis (3,7)	José (8,6)	Ángela (5,8)
Centroide 1 (3,7)	$3\sqrt{5}$	**0**	$\sqrt{26}$	$\sqrt{\mathbf{5}}$
Centroide 2 (22/3,6)	$\frac{\sqrt{\mathbf{61}}}{\mathbf{3}}$	$\frac{\sqrt{178}}{3}$	$\frac{\mathbf{2}}{\mathbf{3}}$	$\frac{\sqrt{85}}{3}$

Los clusters ahora están formados por:

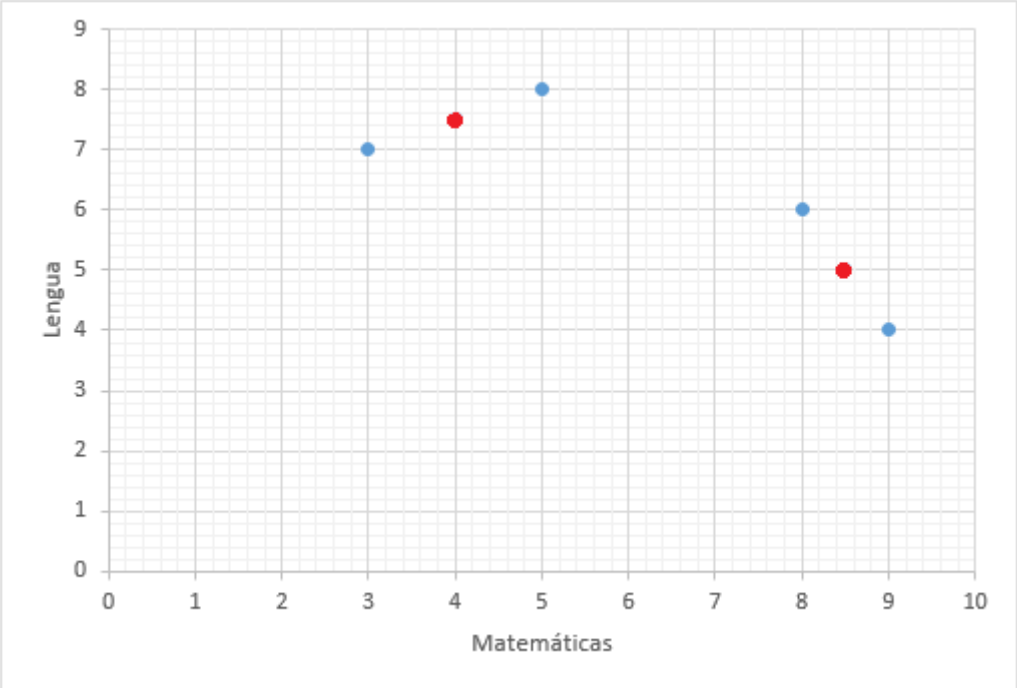
Cluster1 = {Luis, José}
Cluster2 = {Juan, Ángela}

Iteración 3:

Cálculo de nuevos centroides:

Centroide1: ((3+5)/2, (7+8)/2) = (4, 15/2)

Centroide2: ((9+8)/2, (4+6)/2) = (17/2, 5)



Cálculo de distancias y centroide más cercano:

	Juan (9,4)	Luis (3,7)	José (8,6)	Ángela (5,8)
Centroide 1 (4, 15/2)	$\frac{\sqrt{149}}{2}$	$\frac{\sqrt{5}}{2}$	$\frac{\sqrt{73}}{2}$	$\frac{\sqrt{5}}{2}$
Centroide 2 (17/2, 5)	$\frac{\sqrt{5}}{2}$	$\frac{\sqrt{137}}{2}$	$\frac{\sqrt{5}}{2}$	$\frac{\sqrt{85}}{3}$

Los clusters ahora son:

Cluster1 = {Luis, José}
Cluster2 = {Juan, Ángela}

Como los clusters son similares a la iteración anterior, el algoritmo finaliza.

5.3.1.2. Opciones principales

Número de clústeres: Número de clústeres generados.

Función de distancia: Específica qué función de distancia se va a utilizar para los clústeres de K-means. Estas pueden ser euclidiana, coseno o coseno rápido.

Criterio de división: Criterio de división que se va a utilizar para los clústeres de K-means mediante varianza o tamaño.

5.3.2. O-Cluster

5.3.2.1. Descripción

El algoritmo O-clúster de Oracle identifica las agrupaciones que se producen de forma natural en una población de datos. La agrupación en clústeres de partición ortogonal (O-clúster) es un algoritmo de agrupación en clústeres propiedad de Oracle que crea un modelo de agrupación en clústeres jerárquica basado en la cuadrícula, es decir, crea particiones de eje paralelo (ortogonal) en el espacio del atributo de entrada.

El algoritmo funciona de forma recursiva. La estructura jerárquica resultante representa una cuadrícula irregular que forma un mosaico de clústeres en el espacio del atributo.

El algoritmo O-clúster gestiona atributos numéricos y categóricos, y Oracle Data Miner selecciona de forma automática las mejores definiciones de clúster. ODM proporciona

información detallada, reglas y valores centroides del clúster, y se puede utilizar para puntuar una población en relación con su pertenencia a un clúster.

5.3.2.2. Opciones principales

Número de Clusters: Número de clusters generados.

Sensibilidad: Establece una fracción que especifica la densidad máxima necesaria para separar un nuevo clúster. La fracción está relacionada con la densidad uniforme global.

5.4. Ejemplo aplicado a Dataset de la Universidad de Irvine

Para este ejemplo se ha seleccionado un dataset que contiene diferentes medidas del consumo eléctrico de un hogar cada minuto del día durante casi cinco años (http://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption#). Para agilizar el tiempo de computación y centrar el estudio en un mes dado, se ha seleccionado todas las medidas de Junio del año 2006. Se han eliminado también las medidas que contenían algún tipo de valor nulo, quedando finalmente un total de 43158 mediciones.

En la siguiente tabla podemos ver las medidas que han sido almacenadas en el dataset.

ATRIBUTO	SIGNIFICADO	POSIBLES VALORES
Date_time	Fecha y hora	Fecha y hora
global_active_power	Potencia consumida en la instalación eléctrica (Vatios)	Valor continuo
global_reactive_power	No es la potencia realmente consumida en la instalación, ya que no produce trabajo útil. Aparece en una instalación eléctrica en la que existen bobinas o condensadores, y es necesaria para crear campos magnéticos y eléctricos en dichos componentes. (Vatios)	Valor continuo
Voltaje	Voltaje (media durante un minuto). (Voltios)	Valor continuo
global_intensity	Intensidad (media durante un minuto). (Amperios)	Valor continuo
sub_metering_1	Vatio/hora de energía activa en la cocina.	Valor continuo
sub_metering_2	Vatio/hora de energía activa en el lavadero.	Valor continuo
sub_metering_3	Vatio/hora de energía activa del aire acondicionado y calentador eléctrico.	Valor continuo

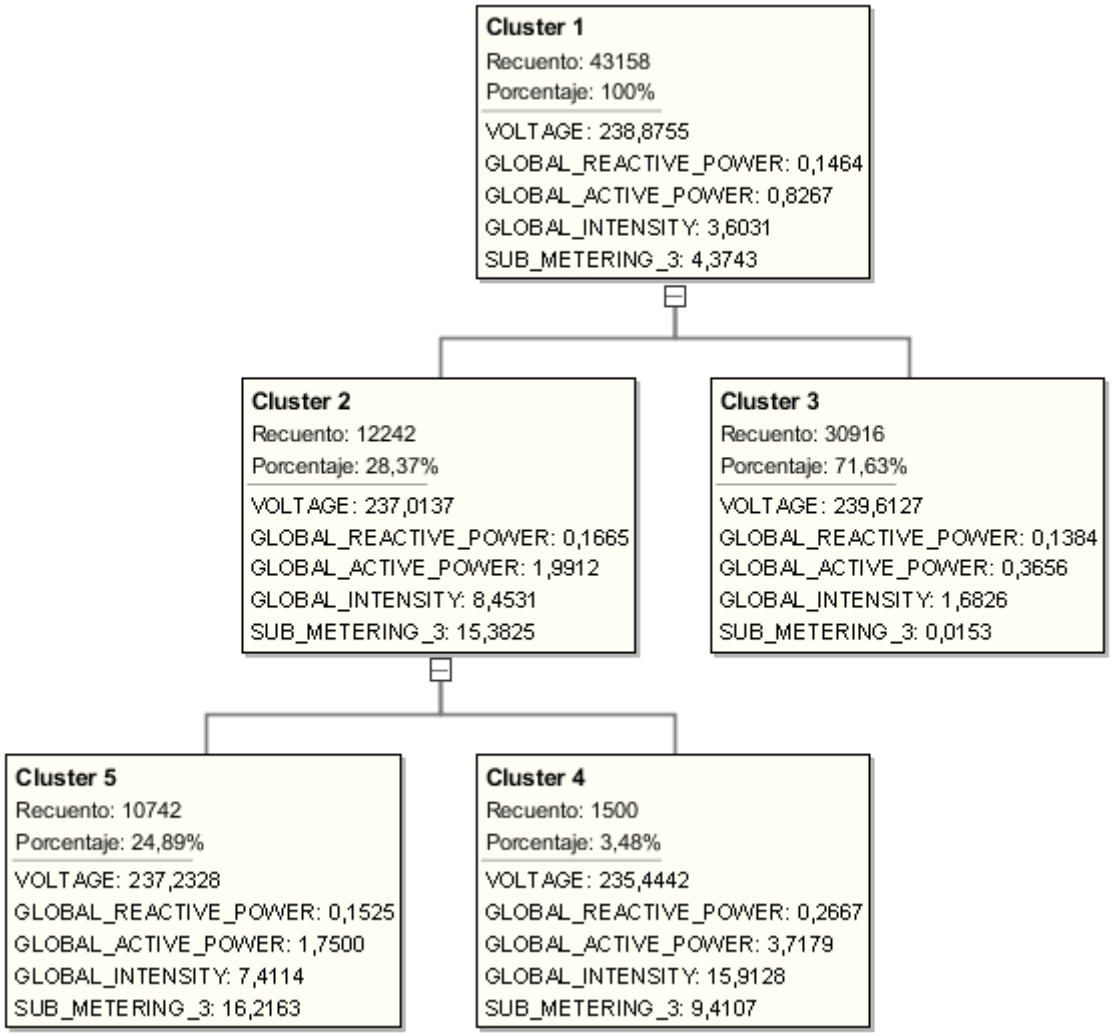
En la siguiente tabla podemos ver un ejemplo de la información que contiene el dataset:

Date_Time	Global_ active_ power	Global_ reactive_ power	Voltage	global_ intensity	Sub_ metering_1	Sub_ metering_2	Sub_ metering_3
01/06/2007 10:24	1,294	0,062	232,14	5,6	0	2	17
01/06/2007 10:25	1,294	0,06	232,11	5,6	0	1	16
01/06/2007 10:26	1,298	0,06	231,86	5,6	0	1	17
01/06/2007 10:27	1,292	0,06	231,41	5,6	0	2	16
01/06/2007 10:28	1,296	0,06	231,83	5,6	0	1	16
01/06/2007 10:29	1,348	0,112	232,18	5,8	0	2	17

Para crear los clusters, añadimos a nuestro proyecto el nodo “origen de datos”, al cual le mostramos donde se encuentra almacenada nuestra información. También añadimos el nodo “creación de clúster”. Los unimos e indicamos en las opciones del nodo “creación de clúster” que queremos usar los dos algoritmos (k-means y o-cluster). Dejamos las configuraciones de los algoritmos por defecto, exceptuando el número de clusters, donde vamos a indicarle que solo queremos tres. Una vez hecho esto, procedemos a ejecutar el nodo “creación de clúster”.

Para ver los clusters que ha generado cada algoritmo solo tenemos que pulsar con el botón secundario sobre el nodo “creación del clúster” y seleccionar el modelo que queremos ver.

Seleccionando el modelo generado por el algoritmo K-MEANS la primera información que nos aparece es la siguiente:



Como se puede ver, se obtiene un árbol el cual está compuesto por 5 clusters.

Los nodos hoja son realmente los clusters en los que posteriormente será dividida nuestra información, en este caso 3, 4 y 5.

Los nodos restantes son clusters intermedios formados por los nodos hoja, por ejemplo los nodos 4 y 5 son subconjuntos del nodo 2.

En cada nodo se muestra el número de elementos que pertenece a él (Recuento) y el porcentaje de la muestra que representan estos elementos. También se indica la media de cada uno de los atributos, por ejemplo en el nodo 5 la media de los voltajes referentes a los elementos que contiene ese clúster es de 237,2328 Voltios.

Si pulsamos sobre cualquiera de los nodos hoja podemos ver cuáles son las reglas usadas para determinar a qué nodo (clúster) pertenece cada elemento.

En la siguiente tabla se pueden observar estas reglas y con cual nodo corresponden:

Clúster	Regla
3	Si 0 <= SUB_METERING_3 <= 2 Y 0,4 <= GLOBAL_INTENSITY <= 3,62 Y 0,082 <= GLOBAL_ACTIVE_POWER <= 0,8352 Y 234,648 <= VOLTAGE <= 244,708 Y 0 <= SUB_METERING_1 <= 7,5 Y 0 <= SUB_METERING_2 <= 7,3 Y 0 <= GLOBAL_REACTIVE_POWER <= 0,3444
4	Si 22,5 <= SUB_METERING_1 <= 45 Y 10,06 <= GLOBAL_INTENSITY <= 26,16 Y 2,3416 <= GLOBAL_ACTIVE_POWER <= 6,1076 Y 228,612 <= VOLTAGE <= 240,684 Y 0 <= GLOBAL_REACTIVE_POWER <= 0,574 Y 0 <= SUB_METERING_2 <= 7,3 Y 0 <= SUB_METERING_3 <= 20
5	Si 3,62 <= GLOBAL_INTENSITY <= 13,28 Y 0,8352 <= GLOBAL_ACTIVE_POWER <= 3,0948 Y 230,624 <= VOLTAGE <= 242,696 Y 0 <= SUB_METERING_1 <= 7,5 Y 0 <= SUB_METERING_2 <= 7,3 Y 0 <= SUB_METERING_3 <= 20 Y 0 <= GLOBAL_REACTIVE_POWER <= 0,3444

Si seleccionamos un registro aleatoriamente como el siguiente, podemos calcular a que clúster pertenece.

DATE_TIME	01/06/2007 12:15:00
GLOBAL_ACTIVE_POWER	1,406
GLOBAL_INTENSITY	6
GLOBAL_REACTIVE_POWER	0,19
SUB_METERING_1	0
SUB_METERING_2	2
SUB_METERING_3	17
VOLTAGE	233,92

Si comprobamos cuales de las reglas cumple, vemos que el registro pertenece al cluster número 5.

Clúster 5

```
Si  3,62 <= GLOBAL_INTENSITY (6) <= 13,28
Y 0,8352 <= GLOBAL_ACTIVE_POWER (1,406) <= 3,0948
Y 230,624 <= VOLTAGE (233,92) <= 242,696
Y 0 <= SUB_METERING_1 (0) <= 7,5
Y 0 <= SUB_METERING_2 (2) <= 7,3
Y 0 <= SUB_METERING_3 (17) <= 20
Y 0 <= GLOBAL_REACTIVE_POWER (0,19) <= 0,3444
```

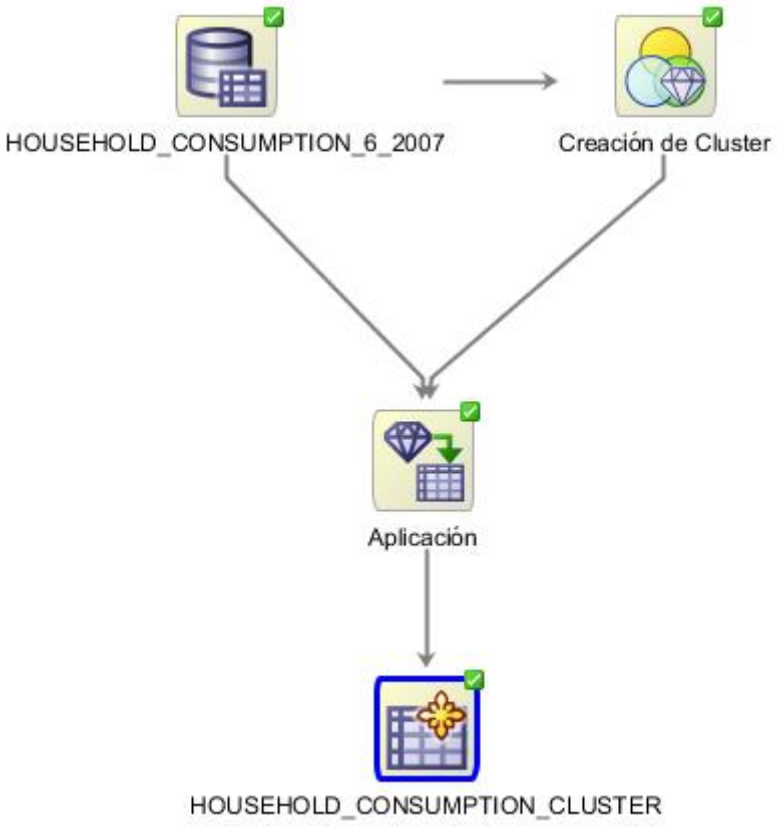
Ahora que conocemos los clusters podemos estudiar toda la información que nos ofrece el modelo para intentar determinar (si es posible) que significado puede tener

cada uno. Personalmente me he fijado en el parámetro intensidad, que nos indica la intensidad global de corriente que es usada por la red.

Si observamos la condición de la intensidad en cada uno de los clusters podemos llegar a la conclusión siguiente:

En el clúster 3 la intensidad se sitúa entre 0,4 y 3,62 Amperios, lo que indica que prácticamente no se está haciendo uso de los aparatos eléctricos. En el clúster 5 la intensidad se mueve entre 3,62 y 12,28 Amperios, lo cual podríamos considerar como un uso medio de la instalación. Y finalmente el clúster 4 donde la intensidad en encuentra entre 10,06 y 26,16 Amperios, muestra un uso intensivo de la red, porque por ejemplo se esté usando el aire acondicionado o el calentador de agua eléctrico.

Una vez que tenemos nuestro nodo de clustering preparado, procedemos a aplicarlo a nuestros datos, para saber a qué clúster pertenece cada elemento.



Los resultados los vamos a almacenar en una tabla llamada HOUSEHOLD_CONSUMPTION_CLUSTER donde para simplificar solo vamos a mostrar la fecha y la hora junto con el clúster al que pertenece y la probabilidad de que sea cierto.

Sabiendo ya lo que significa que un elemento pertenezca a un clúster u otro, en este caso, podemos por ejemplo ver las horas en las cuales se hace mayor o menor uso de la instalación eléctrica.

Aquí se puede observar un extracto de la tabla:

DATE_TIME	CLUS_KM_1_12_CLID	CLUS_KM_1_12_PROB
01/06/2007 20:37	0,7965	5
01/06/2007 20:38	0,8053	5
01/06/2007 20:39	0,7694	5
02/06/2007 10:30	0,7409	3
02/06/2007 10:31	0,7399	3
02/06/2007 10:32	0,7411	3
02/06/2007 11:04	0,998	4
02/06/2007 11:05	0,9975	4
02/06/2007 11:06	0,9974	4

En la tabla podemos ver que las 8 de la tarde pertenece al clúster número 5, lo que nos indica que se está haciendo un uso medio de la red.

Las 10 de la mañana pertenece al clúster número 3 por lo que no se está haciendo uso de la red.

Las 11 de la mañana pertenece al clúster número 4 lo que nos indica que se está haciendo un uso intensivo de la red.

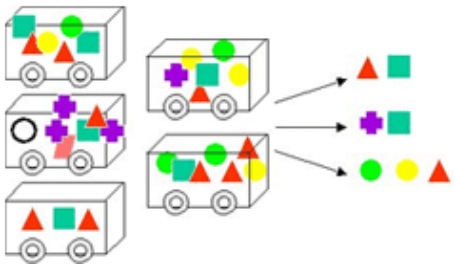
5.5. Ejemplo aplicado a Dataset de la Universidad de Irvine mediante PL/SQL

```
/* Creación de la tabla que contendrá las opciones del modelo */
CREATE TABLE CONSUMPTION_CLUS_SET (
    SETTING_NAME VARCHAR(30),
    SETTING_VALUE VARCHAR(4000)
  );


/* Inserción de las opciones */
BEGIN

  /* Algoritmo: K-MEANS */
  INSERT INTO CONSUMPTION_CLUS_SET (SETTING_NAME, SETTING_VALUE) VALUES
  (SYS.dbms_data_mining.ALGO_NAME, SYS.dbms_data_mining.ALGO_KMEANS);

  /* Número de clusters: 3 */
  INSERT INTO CONSUMPTION_CLUS_SET (SETTING_NAME, SETTING_VALUE) VALUES
  (SYS.dbms_data_mining.CLUS_NUM_CLUSTERS, 3);

  /* Preparación automática de los datos */
  INSERT INTO CONSUMPTION_CLUS_SET (SETTING_NAME, SETTING_VALUE) VALUES
  (SYS.dbms_data_mining.PREP_AUTO, SYS.dbms_data_mining.PREP_AUTO_ON);

END;
```

```
/* Creación del Modelo a partir de los datos situados en la
   tabla HOUSEHOLD_CONSUMPTION_6_2007 */
BEGIN
  SYS.dbms_data_mining.CREATE_MODEL(
    MODEL_NAME => 'CONSUMPTION_CLUS_PL',
    MINING_FUNCTION => DBMS_DATA_MINING.CLUSTERING,
    DATA_TABLE_NAME => 'HOUSEHOLD_CONSUMPTION_6_2007',
    CASE_ID_COLUMN_NAME => 'DATE_TIME',
    TARGET_COLUMN_NAME => NULL,
    SETTINGS_TABLE_NAME => 'CONSUMPTION_CLUS_SET'
  );
END;
/

/* Aplicación del modelo a los datos situados en la tabla
   HOUSEHOLD_CONSUMPTION_6_2007 y volcado del resultado en la tabla
   CONSUMPTION_RESULT_PL */
BEGIN
  SYS.DBMS_DATA_MINING.APPLY(
    model_name => 'CONSUMPTION_CLUS_PL',
    data_table_name => 'HOUSEHOLD_CONSUMPTION_6_2007',
    case_id_column_name => 'DATE_TIME',
    result_table_name => 'CONSUMPTION_RESULT_PL'
  );
END;
/
```

6. Técnica de Asociación

6.1. Descripción



La asociación es una técnica de minería de datos que nos permite averiguar hechos que ocurren dentro de los conjuntos de datos. Cuando aplicamos esta técnica, el resultado es un conjunto de reglas de asociación.

Esta reglas tiene este aspecto: {A, B} => {C} ({antecedente} => {consecuente}), esto quiere decir que siempre que se da A y B es muy probable que también se dé C. Para poner en contexto esto, imaginemos que contamos con un conjunto de datos donde se indica que libros extraen los alumnos de la biblioteca de ETSI Informática de Málaga. Algunas de las posibles reglas que podríamos obtener tendrían un aspecto parecido a estas:

{“Java 8, los fundamentos del lenguaje Java”, “Fundamental 2d game programing with Java”} => {“Programación en Android con Java”}

Esta regla nos hace ver que la mayoría de los alumnos que han pedido un préstamo de los libros “Java 8, los fundamentos del lenguaje Java” y “Fundamental 2d game programing with Java” también se han llevado el libro “Programación en Android con Java”.

{“Dispositivos Electrónicos”, “Análisis de circuitos electrónicos”} => {“Problemas Resueltos de circuitos electrónicos”}

En esta observamos que los alumnos que han pedido un préstamo de los libros “Dispositivos Electrónicos” y “Análisis de circuitos electrónicos” también se han llevado en la mayoría de los casos “Problemas Resueltos de circuitos electrónicos”.

A partir de este análisis, por ejemplo, podríamos obtener como conclusión que tener en stock él mismo número de volúmenes de los libros indicados en la reglas sería buena idea, ya que así, casi siempre un alumno va a poder llevarse los tres libros juntos.

6.2. Algoritmos de Asociación

6.2.1. A priori

6.2.1.1. Descripción

El algoritmo Apriori encuentra reglas de asociación en los datos. Por ejemplo, "si un cliente de una carpintería ha comprado madera contrachapada y tacos, hay un 89% de probabilidad de que también compre cola blanca". El algoritmo divide su ejecución en dos partes:

1- Encontrar todas las combinaciones de elementos, denominadas conjuntos de elementos frecuentes, cuyo soporte es superior al soporte mínimo.

2- Utilizar los conjuntos de elementos frecuentes para generar las reglas deseadas. La idea es que si, por ejemplo, ABC y BC son frecuentes, entonces la regla "A implica BC", siempre que el cociente de soporte (ABC) y soporte (BC) sea como mínimo igual de grande que la confianza mínima. Observe que la regla tendrá un soporte mínimo debido a que ABCD es frecuente. La

asociación de Oracle Data Miner solamente admite reglas únicas consecuentes (ABC implica D).

6.2.1.2. *Opciones Principales*

Longitud Máxima de Regla: Número máximo de precondiciones que puede tener una regla.

Porcentaje confianza mínima: Las reglas con confianza mínima menor a la especificada son descartadas.

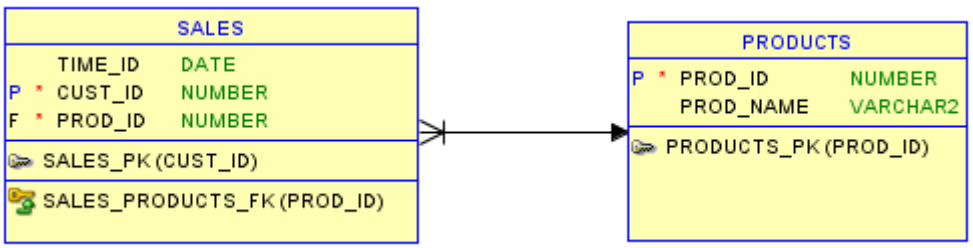
Porcentaje soporte mínimo: porcentaje mínimo para considerar un conjunto de datos frecuente.

6.3. Ejemplo aplicado a Dataset

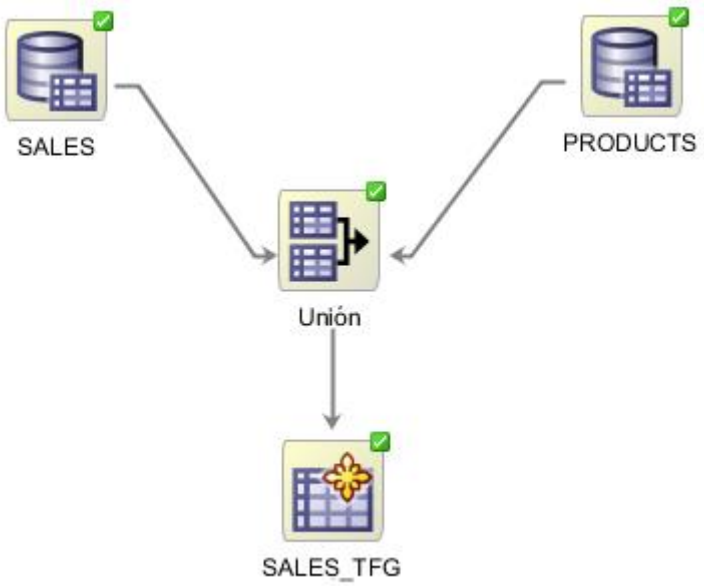
Desgraciadamente no existe ningún dataset en la universidad de Irvine que cumpla con las características necesarias para aplicarle la técnica de asociación. Por ello, vamos a elegir un dataset que ofrece Oracle en uno de sus esquemas. Además dado que la compresión de este es muy simple e intuitiva, obviaremos el apartado de "Ejemplo simple".

Este dataset se encuentra situado en el esquema SH de Oracle 11g y su nombre es SALES. Este contiene una lista de los productos adquiridos por los clientes de un establecimiento, en total 918843 elementos. Para facilitar la compresión del dataset vamos a realizar una unión (join) de las tablas SH.SALES y SH.PRODUCTS con el objetivo de tener el nombre del producto en vez de su identificador. Además para simplificar el dataset solo vamos a usar los siguientes campos:

TABLA SALES	TABLA PRODUCTS
CUST_ID, TIME_ID y PROD_ID (para realizar la unión).	PROD_NAME, PROD_ID (para realizar la unión).



La unión la podemos crear mediante el nodo “Unión” indicando en las propiedades, que vamos a realizarla a partir de la clave ID_PROD situada en ambas tablas. También eliminaremos el atributo PROD_ID en el nodo “Crear nueva tabla o vista”.



Quedando finalmente nuestro dataset con el siguiente aspecto:

ATRIBUTO	SIGNIFICADO	POSIBLES VALORES
CUST_ID	Identificador del cliente	Valor Discreto
TIME_ID	Fecha en la que fue realizada la compra	Fecha
PROD_NAME	Nombre del producto	Cadena de texto

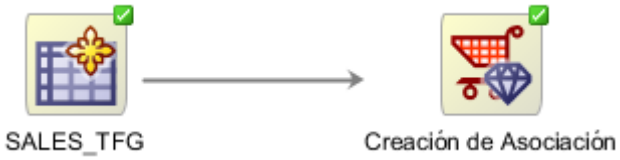
Un ejemplo de la información que contiene es el siguiente:

CUST_ID	TIME_ID	PROD_NAME
2	02/05/1998	Mouse Pad
2	02/05/1998	1.44MB External 3.5" Diskette
6	22/06/2001	Model C9827B Cordless Phone Battery
6	22/06/2001	Model K8822S Cordless Phone Battery
20	17/08/1999	Envoy External 6X CD-ROM
20	17/08/1999	SIMM- 16MB PCMCIAII card
20	17/08/1999	Multimedia speakers- 3" cones
20	17/08/1999	Unix/Windows 1-user pack

Como podemos observar, cada compra de un producto ha sido insertada de forma individual, por lo que para saber que llevaba en el “carrito” cierto día un cliente, deberíamos agrupar por fecha e identificador de cliente. Por ejemplo, el cliente con identificador 2, el día 2/5/1989 compró un Mouse Pad y 1.44MB External 3.5" Diskette.

Ahora que tenemos un dataset comprensible podemos proceder a obtener las reglas de asociación que en él subyacen.

Para poder aplicar el nodo “Creación de asociación”, necesitamos indicar en sus propiedades el ID de la transacción (en este caso la transacción se refiere a la compra del usuario) y el identificador del elemento del cual queremos las reglas de asociación. Para nosotros el identificador de una transacción está compuesto por el identificador de usuario y la fecha de compra. Y el identificador del elemento es el nombre del producto.



Una vez ejecutado el nodo “Creación de asociación” procedemos a examinar el modelo generado.

En la primera pestaña, llamada “Reglas”, tal y como su nombre índica tenemos todas las reglas encontradas por el algoritmo.

Aquí podemos ver las cinco primeras:

ID	Antecedente	Consecuente	Medida de Eficacia	Confianza (%)	Soporte (%)	Recuento de Elementos
9704	128MB Memory Card AND Comic Book Heroes AND 256MB Memory Card	Fly Fishing	29,123	70,866	1,069	3
11589	Bounce AND Comic Book Heroes AND 256MB Memory Card	Fly Fishing	28,739	69,93	1,053	3
9716	128MB Memory Card AND Martial Arts Champions AND Comic Book Heroes	Fly Fishing	28,302	68,868	1,1	3
11601	Bounce AND Martial Arts Champions AND Comic Book Heroes	Fly Fishing	27,853	67,774	1,098	3
9701	Fly Fishing AND 256MB Memory Card AND 128MB Memory Card	Comic Book Heroes	27,826	77,547	1,069	3

El significado de la primera fila es el siguiente:

Las dos primeras columnas representan el **Antecedente** y el **Consecuente** de la regla. Representado de manera convencional quedaría de la siguiente forma:

{128MB Memory Card AND Comic Book Heroes AND 256MB Memory Card} => {Fly Fishing}

Es decir, la mayoría de clientes que han adquirido “128MB Memory Card”, “Comic Book Heroes” y “256MB Memory Card” también han adquirido “Fly Fishing”.

El **Soporte** nos indica, qué tanto por ciento de veces aparecen juntos los elementos del antecedente en las distintas transacciones. Es decir, en nuestro caso, cuántas veces “128MB Memory Card”, “Comic Book Heroes” y “256MB Memory Card” aparecen en una compra.

El soporte para la primera regla es de 1,069% aunque nos parezca un valor bastante pequeño, hay que tener en cuenta que el dataset está formado por 918843 elementos que agrupados en transacciones (*select count(*) from (select count(*) from sh.sales group by cust_id,time_id)*) hacen un total de 143139 transacciones, por lo que el conjunto de datos del antecedente aparece en 1530 de las transacciones (compras).

$$Soporte = \frac{Número\ de\ transacciones\ en\ la\ que\ aparecen\ juntos\ lo\ elementos\ del\ antecedente}{Número\ total\ del\ transacciones}$$

La **Confianza** muestra el tanto por ciento de veces que se ha dado la regla.

En nuestro caso la regla {128MB Memory Card AND Comic Book Heroes AND 256MB Memory Card} => {Fly Fishing} se ha cumplido el 70,866% de la veces. Es decir, de la muestra que hemos usado para crear las reglas de asociación, el 70,866% de la veces que un cliente ha comprado “128MB Memory Card”, “Comic Book Heroes” y “256MB Memory Card” también ha comprado “Fly Fishing”.

$$Confianza(Antecedente => Consecuente) = \frac{Soporte\ (Antecedente\ U\ Consecuente)}{Soporte(Antecedente)}$$

La **Medida de Eficacia**:

$$Medida\ de\ Eficia\ (Antecente => Consecuente) = \frac{Soporte(Antecedente\ U\ Consecuente)}{Soporte(Antecedente) x Soporte(Consecuente)}$$

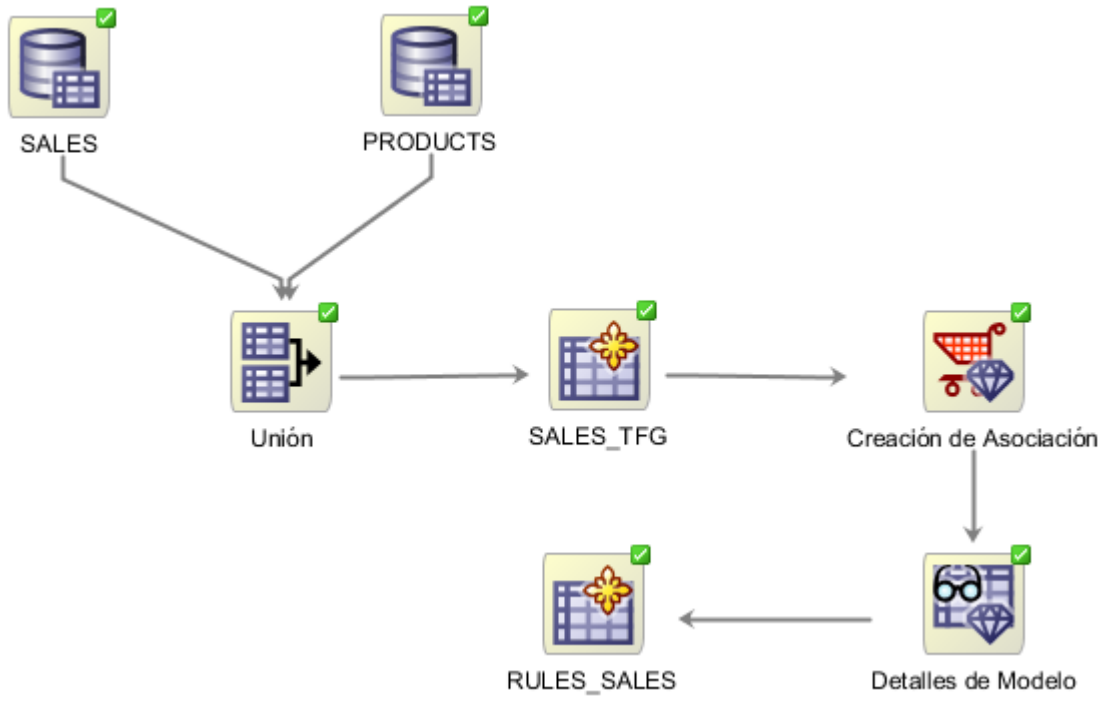
Finalmente tenemos el **Recuento de elementos**, el cual nos indica el número de elementos de los que está compuesto el antecedente.

Como sabemos el algoritmo A priori, en su primera parte de ejecución genera los "Conjuntos de elementos frecuentes". Estos conjuntos generados los podemos ver en la pestaña "Juego de Elementos". Si accedemos a ella nos aparece algo como esto:

ID	Elementos	Soporte (%)	Recuento de Elementos
717	3 1/2" Bulk diskettes, Box of 50, Model SM26273 Black Ink Cartridge	5,06	2
408	Internal 8X CD-ROM, Envoy External 8X CD-ROM	5,02	2
3817	Music CD-R, CD-R, Professional Grade, Pack of 10, CD-R with Jewel Cases, pack OF 12, CD-RW, High Speed Pack of 5	4,981	4
2930	Model A3827H Black Image Cartridge, Model K3822L Cordless Phone Battery, Model C9827B Cordless Phone Battery	4,97	3

El significado de la primera fila sería que el conjunto con identificador 717, formado por los elementos "3 1/2" Bulk diskettes, Box of 50" y "Model SM26273 Black Ink Cartridge" tiene un soporte del 5,06%. Lo que quiere decir que este conjunto de elementos aparece en el 5,06% de las transacciones (compras).

Finalmente, si queremos hacer persistentes las reglas generadas, las podemos almacenar en una nueva tabla añadiendo el nodo "Detalles del Modelo" y el nodo "Crear tabla o vista" quedando el esquema con el siguiente aspecto:



El contenido de la tabla generada es similar a lo mostrado en la pestaña "Reglas".

6.4. Ejemplo aplicado a Dataset mediante PL/SQL

```
/* Crear una vista donde identifiquemos cada transacción (compra) con la clave
compuesta
   (cust_id + time_id)*/
CREATE VIEW SALES_TFG_PK AS (SELECT CUST_ID||TIME_ID TRANSACTION_ID, PROD_NAME FROM
SALES_TFG);



/* Creación de la tabla que contendrá las opciones del algoritmo */
CREATE TABLE sales_class_settings
(setting_name VARCHAR2(30),setting_value VARCHAR2(4000));


/* Inserción de las opciones */
BEGIN
  /* Algoritmo: Apriori */
  INSERT INTO sales_class_settings
  (setting_name, setting_value) VALUES
  (dbms_data_mining.algo_name, dbms_data_mining.algo_apriori_association_rules);

  /* Desactivar preparación automática de los datos */
  INSERT INTO sales_class_settings
  (setting_name, setting_value) VALUES
  (dbms_data_mining.prep_auto, dbms_data_mining.prep_auto_off);

  /* Identificador el elemento del cual queremos obtener las reglas */
  INSERT INTO sales_class_settings
  (setting_name, setting_value) VALUES
  (dbms_data_mining.odms_item_id_column_name, 'PROD_NAME');

  /* Soporte mínimo */
  INSERT INTO sales_class_settings
  (setting_name, setting_value) VALUES
  (dbms_data_mining.asso_min_support, 0.01);
END;
/


/* Creación de reglas de asociación a partir de los datos situados en la
   tabla SALES_TFG */
BEGIN

  SYS.DBMS_DATA_MINING.CREATE_MODEL(
    model_name => 'SALES_CLASS_APRIORI_PL',
    mining_function => dbms_data_mining.association,
    data_table_name => 'SALES_TFG_PK',
    case_id_column_name => 'TRANSACTION_ID',
    target_column_name => NULL,
    settings_table_name => 'sales_class_settings'
  );

  END;
  /

 /* Obtención de los conjuntos de elementos frecuentes, junto con su soporte */
 SELECT ITEMSET_ID, ITEMS, SUPPORT, NUMBER_OF_ITEMS FROM
```

```
TABLE(DBMS_DATA_MINING.GET_FREQUENT_ITEMSETS('SALES_CLASS_APRIORI_PL'));

/* Obtención de las reglas de asociación junto con su soporte y su confianza */
SELECT RULE_ID, ANTECEDENT, CONSEQUENT, RULE_SUPPORT, RULE_CONFIDENCE
FROM
TABLE(DBMS_DATA_MINING.GET_ASSOCIATION_RULES('SALES_CLASS_APRIORI_PL'));
```

7. Desarrollo prototipo de aplicación

Una vez probada la capacidad de Oracle Data Miner para realizar minería de datos mediante su cliente Sqldeveloper, se ha procedido a realizar el desarrollo de un prototipo de una aplicación con la intención de evaluar la integración de la tecnología en una plataforma cualquiera.

La función principal de la aplicación es ayudar en el diagnóstico de un tumor de mama. La aplicación ofrece al médico el diagnostico (maligno o benigno) y la probabilidad de que este sea cierto a partir de una prueba inicial y poco invasiva como es una citología de la mama.

Para el desarrollo se ha utilizado un dataset de la universidad de Irvine llamado**:** **Breast Cancer Wisconsin (Original)** [https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29].

Podemos ver un extracto del dataset en la siguiente tabla dividida en dos niveles:

Id	Clump thickness	Uniformity of cell size	Uniformity of cell shape	Marginal adhesion	Single epithelial cell size
157	1	1	1	1	2
158	1	1	1	1	2
159	10	5	7	3	3
160	3	1	1	1	2
161	2	1	1	2	2

Id	Bare nuclei	Bland chromatin	Normal nucleoli	Mitoses	Diagnostic
157	1	2	1	1	2
158	1	3	1	1	2
159	7	3	3	8	4
160	1	3	1	1	2
161	1	3	1	1	2

Como observamos los atributos han sido previamente normalízados en una escala de 1 a 10. En la aplicación por simplicidad se ha decidido que los datos se introduzcan

de igual manera, pero si la aplicación fuera para un uso real habría que introducirlos con sus valores normales y posteriormente normalizarlos, o partir de un dataset con los datos sin normalizar. El diagnostico se divide en dos clases identificadas por los números 2 (benigno) y 4 (maligno).

Este dataset es usado por la aplicación para creación de modelos de clasificación usando por defecto el algoritmo “Naive Bayes”.

La aplicación está divida en tres partes bien diferenciadas y con diferentes usuarios. En primer lugar tenemos el médico de la paciente. Este médico es el encargado de enviar la solicitud para realizar una citología y posteriormente de consultar sus resultados y la probabilidad de que estos sean ciertos.

La segunda parte de la aplicación corresponde con la labor de la anatomopatóloga. A ella le llegan las peticiones de las citologías y es la encargada de realizarlas, e introducir los resultados en el sistema. Una vez que introduce los datos, el sistema realiza la predicción del diagnóstico y calcula probabilidad de que este sea cierto.

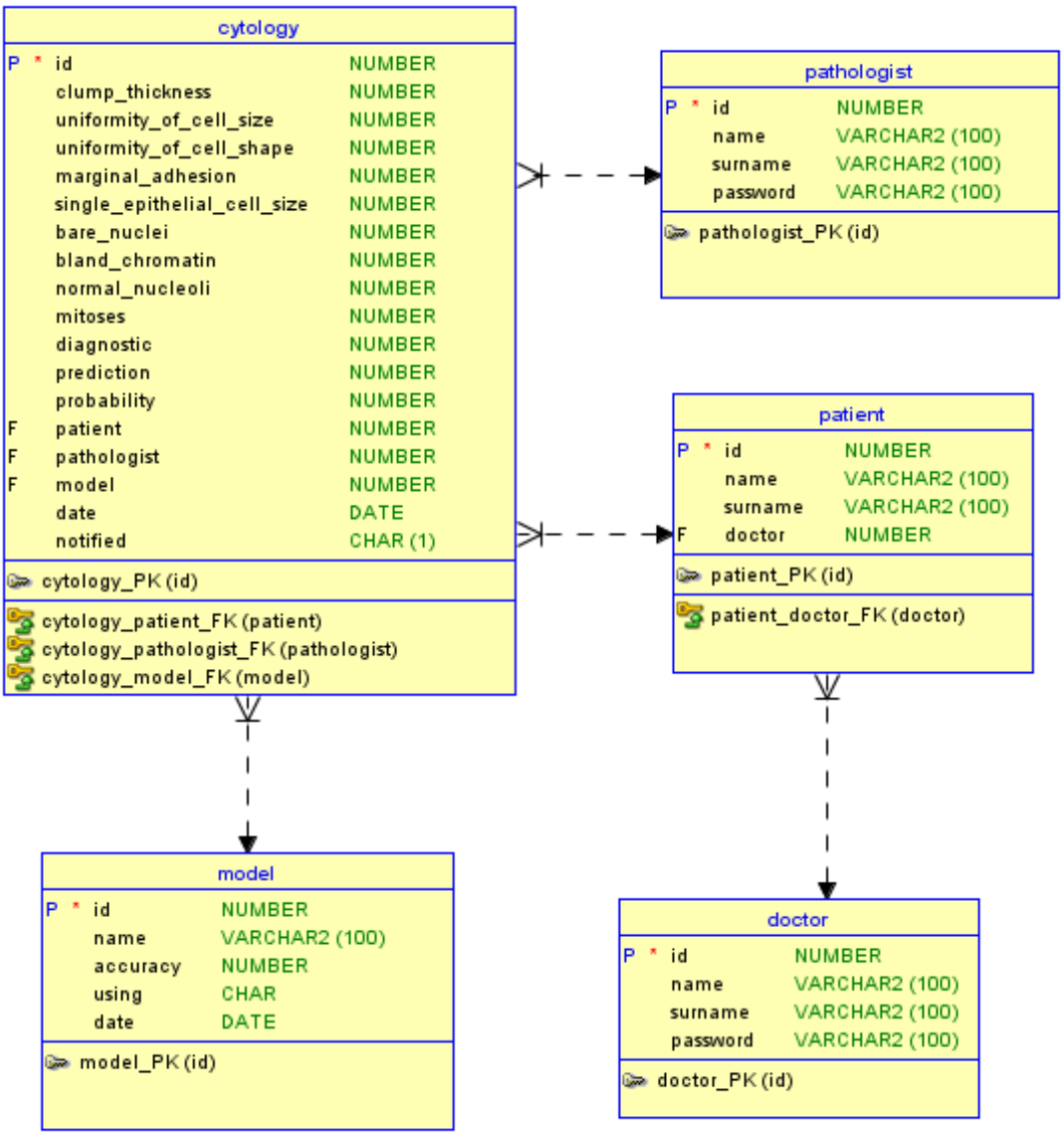
Cuando llegan los resultados al médico, este puede realizar las acciones oportunas, una vez han sido interpretados los datos, como por ejemplo, adelantar a cierta paciente en la lista de espera.

Una vez que el diagnóstico de la paciente es definitivo, el médico lo indica en el sistema. Con ello conseguimos añadir citologías con resultados reales, lo que nos ayudará en un futuro a crear predicciones con mayor precisión.

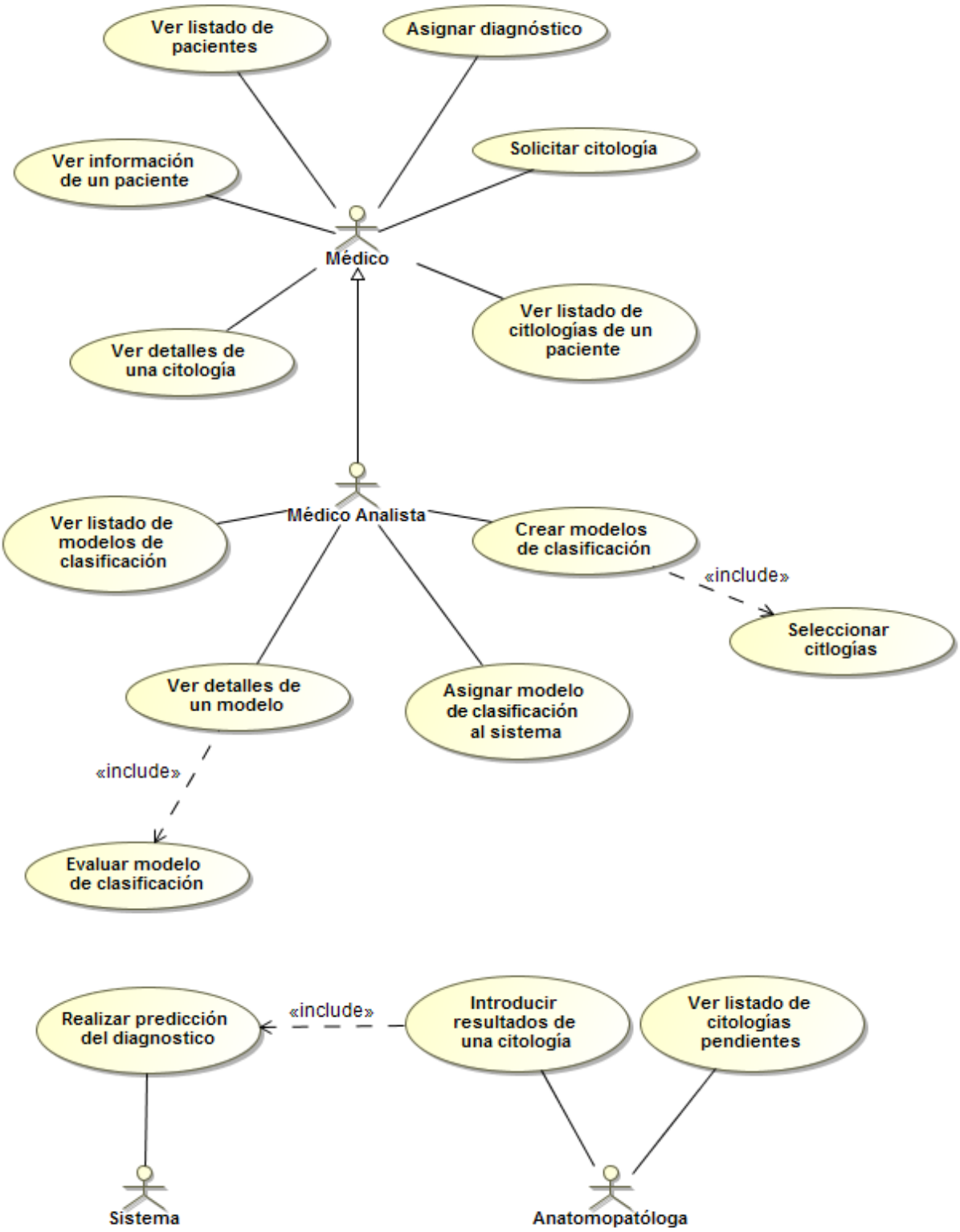
La última parte es la más importante, en ella entra en juego el rol del médico analista. Este médico es el encargado de crear y evaluar los modelos de clasificación que son usados en el sistema para realizar las predicciones cuando la anatomopatóloga introduce los datos. Una vez que tiene un modelo que le ofrece los resultados esperados, puede indicar al sistema que comience a usarlo.

8. Diagramas de la Aplicación

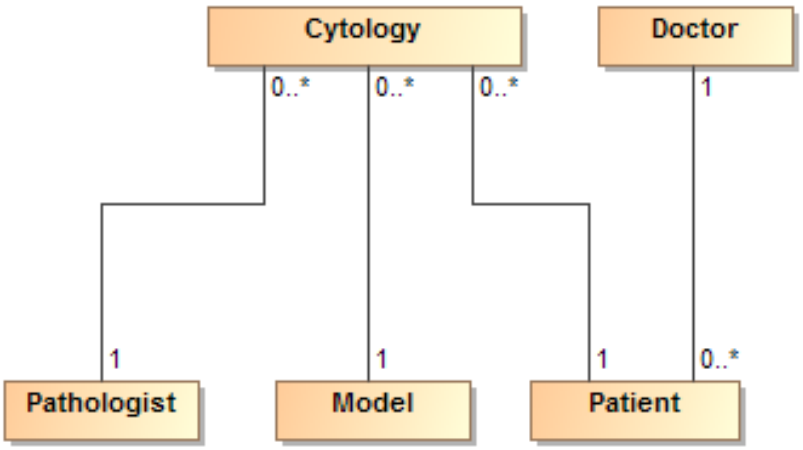
8.1.1. Modelo relacional de la base de datos



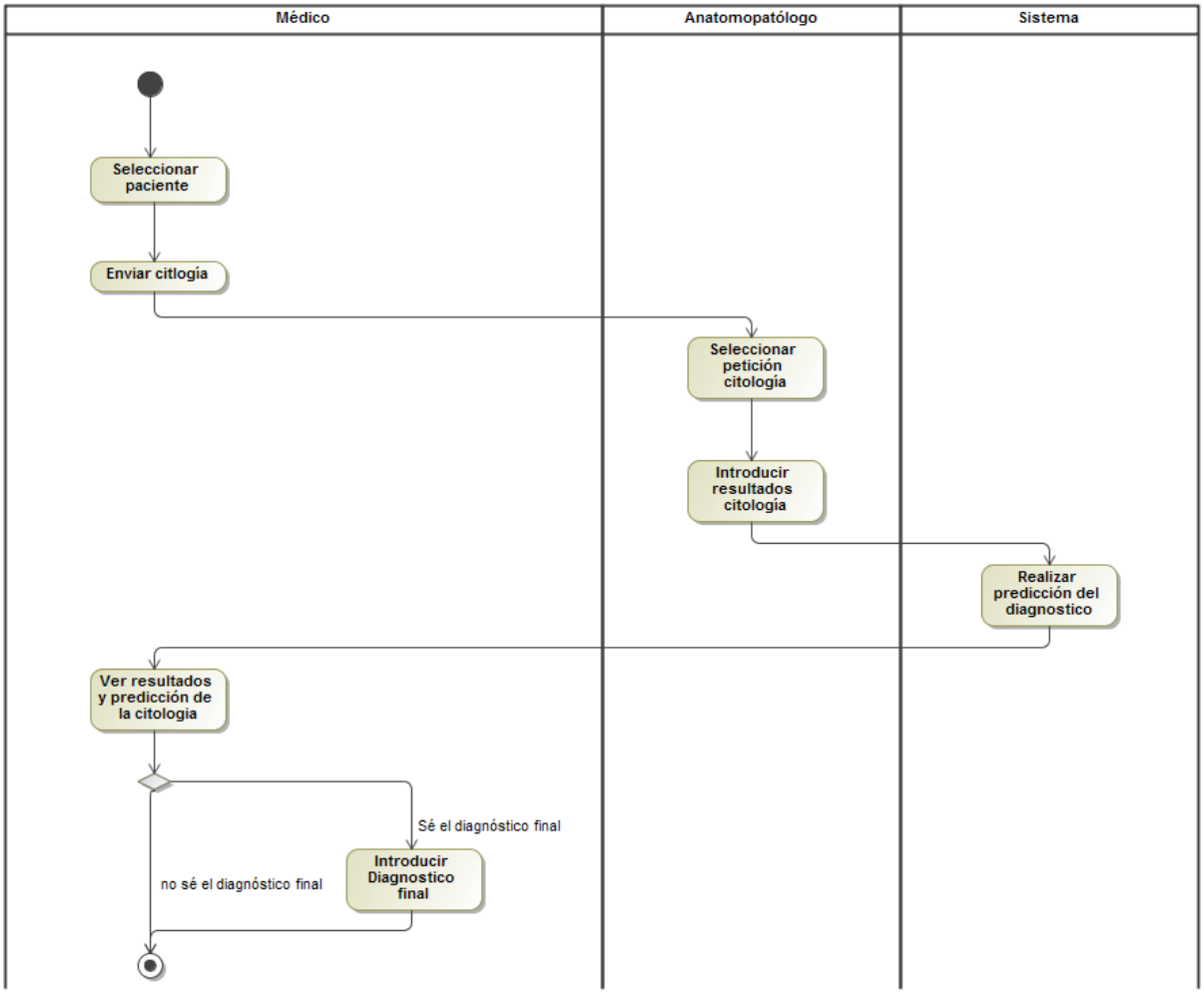
8.1.2. Diagrama de Casos de Uso



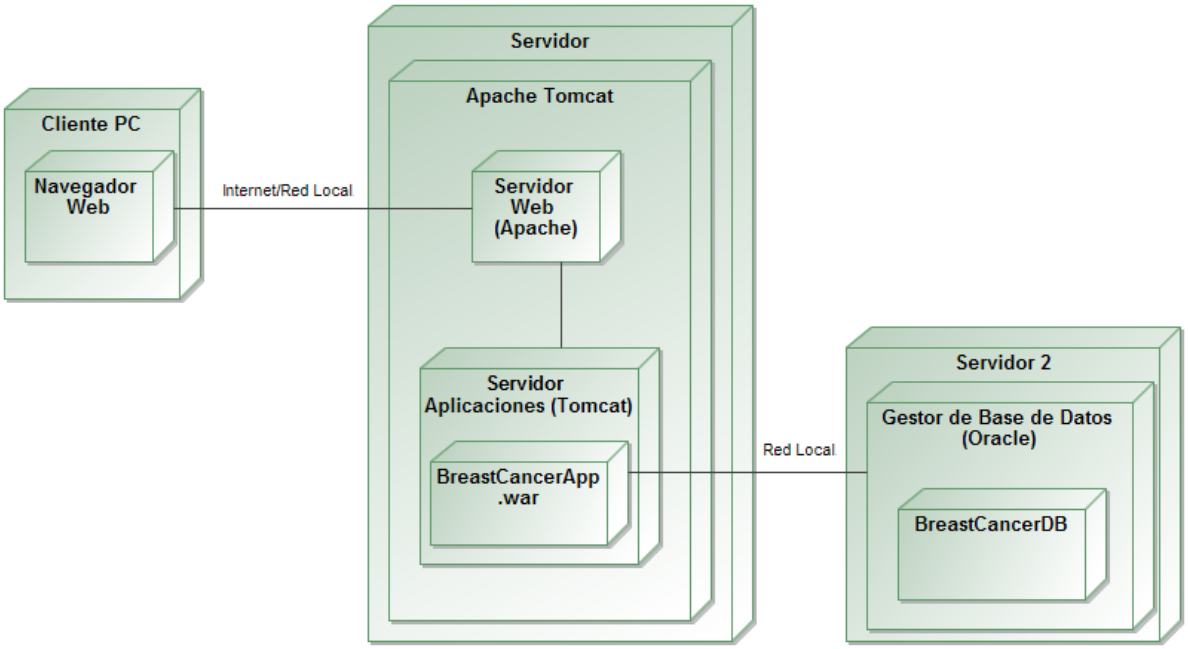
8.1.3. Diagrama de clases del modelo de negocio



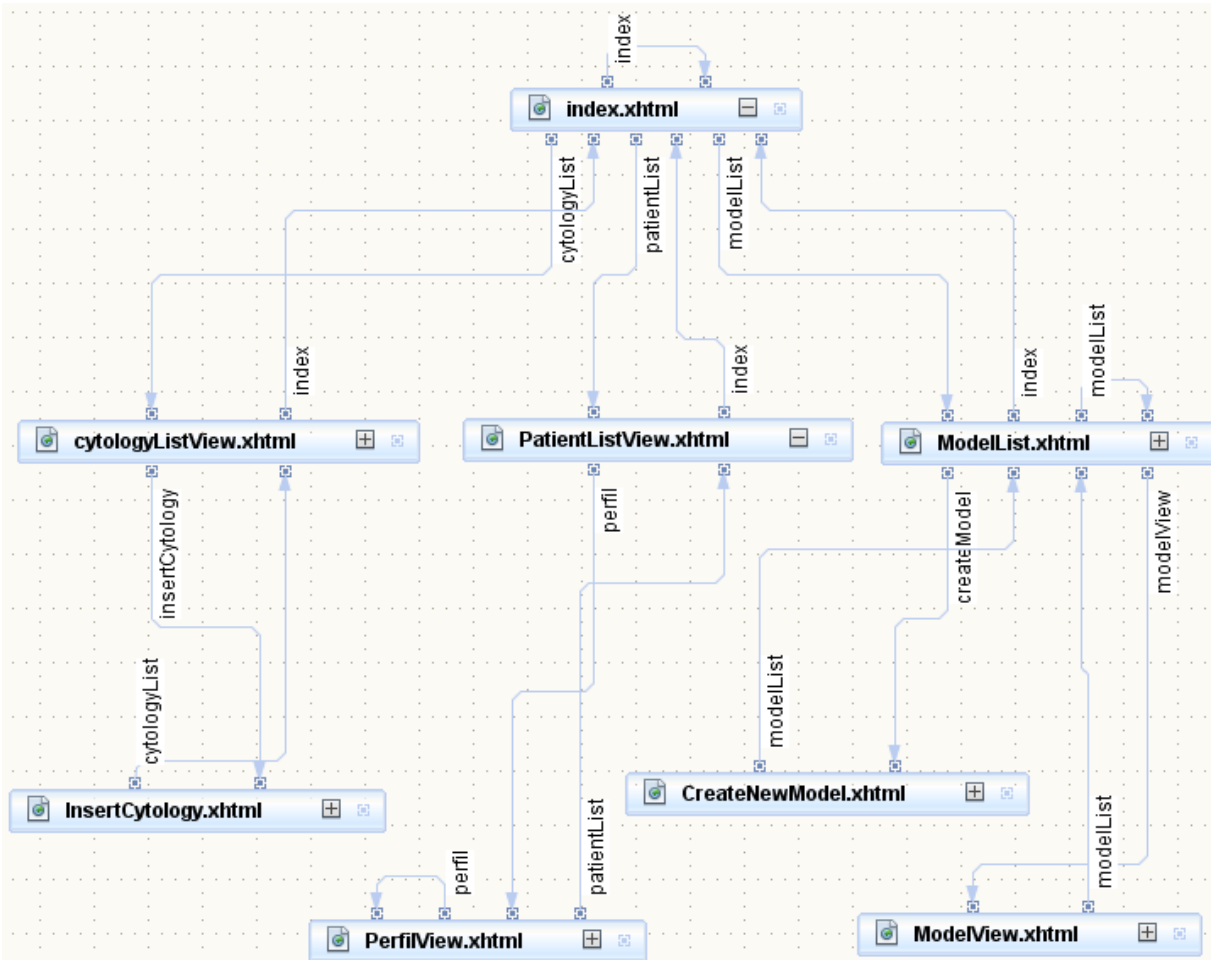
8.1.4. Diagrama de actividad



8.1.5. Diagrama de despliegue



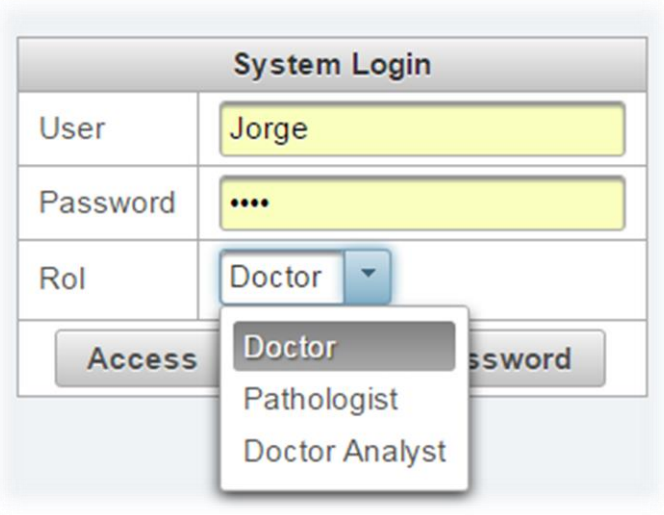
8.1.6. Diagrama de navegación



9. Manual de uso de la aplicación

9.1. Login

La primera pantalla que nos encontramos cuando accedemos a la aplicación es la de Login. El sistema cuenta con 3 roles de usuario: Doctor, Pathologist, Doctor Analyst. Cada uno de ellos nos redirige a su parte de la aplicación especializada.

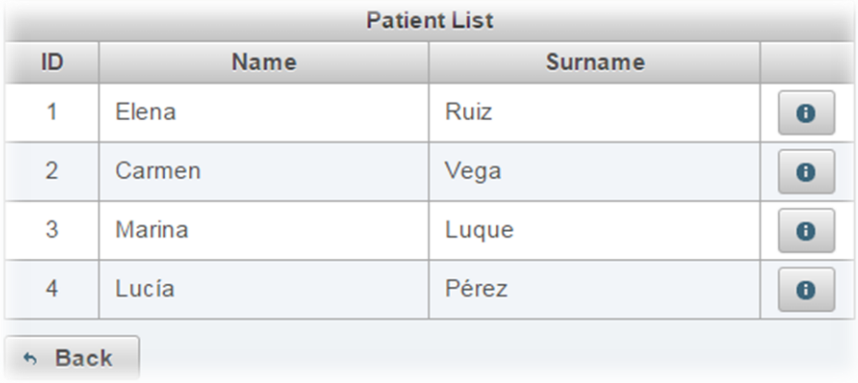


Los usuarios válidos para acceder al sistema son los siguientes:

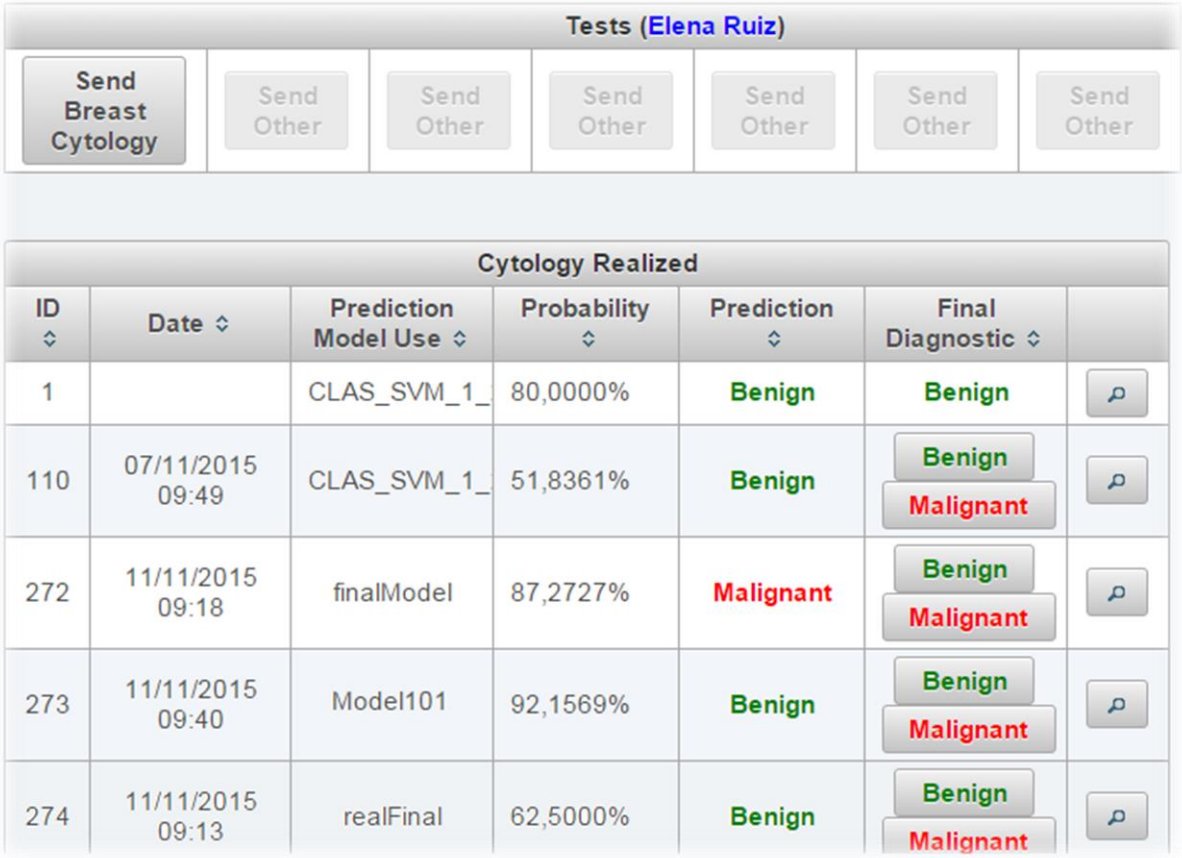
Usuario	Contraseña	Rol
Jorge	1234	Doctor
Lidia	1234	Pathologist
Jorge	1234	Doctor Analyst

9.2. Usuario Doctor

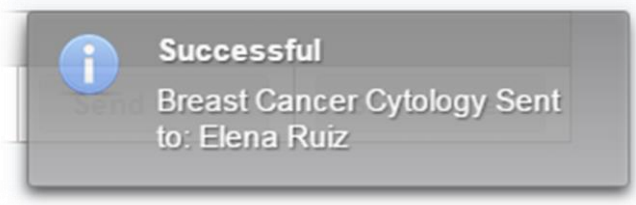
Si accedemos mediante el usuario Jorge con el rol Doctor, el sistema nos muestra el listado de pacientes que tiene asignado el médico.



Cuando pulsamos sobre el icono de información situado a la derecha de cada paciente, el sistema nos muestra en primer lugar las pruebas que se pueden solicitar para este. En nuestro caso, dado que la aplicación es un prototipo, sólo podemos solicitar citologías.



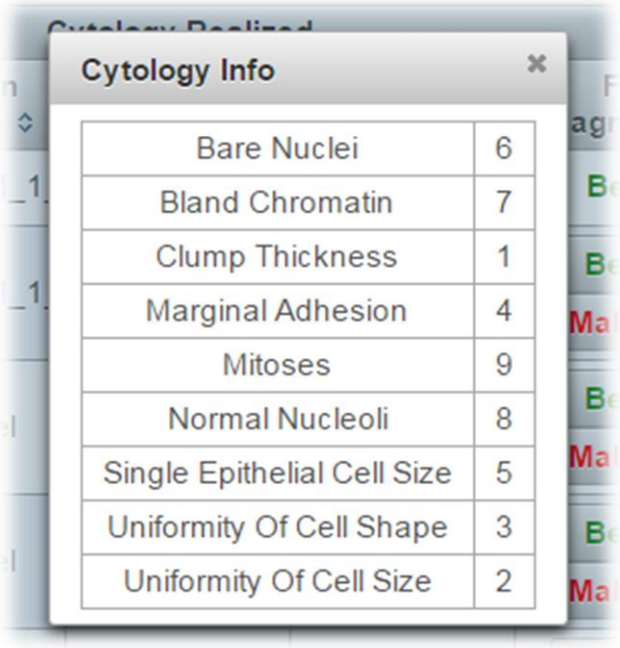
Cuando solicitamos una citología el sistema nos indica si el proceso se ha realizado correctamente mediante el siguiente mensaje:



En segundo lugar, la aplicación nos muestra un listado de las citologías del paciente. De estas podemos distinguir dos tipos, aquellas en las que ya se sabe su diagnóstico final, como por ejemplo la primera y las que todavía no se conocen diagnóstico (todas las demás).

El proceso normal consiste en que el médico recibe los resultados de una citología junto con la predicción del diagnóstico y la probabilidad de que sea cierto. Estos datos ayudan al médico a tomar las decisiones oportunas.

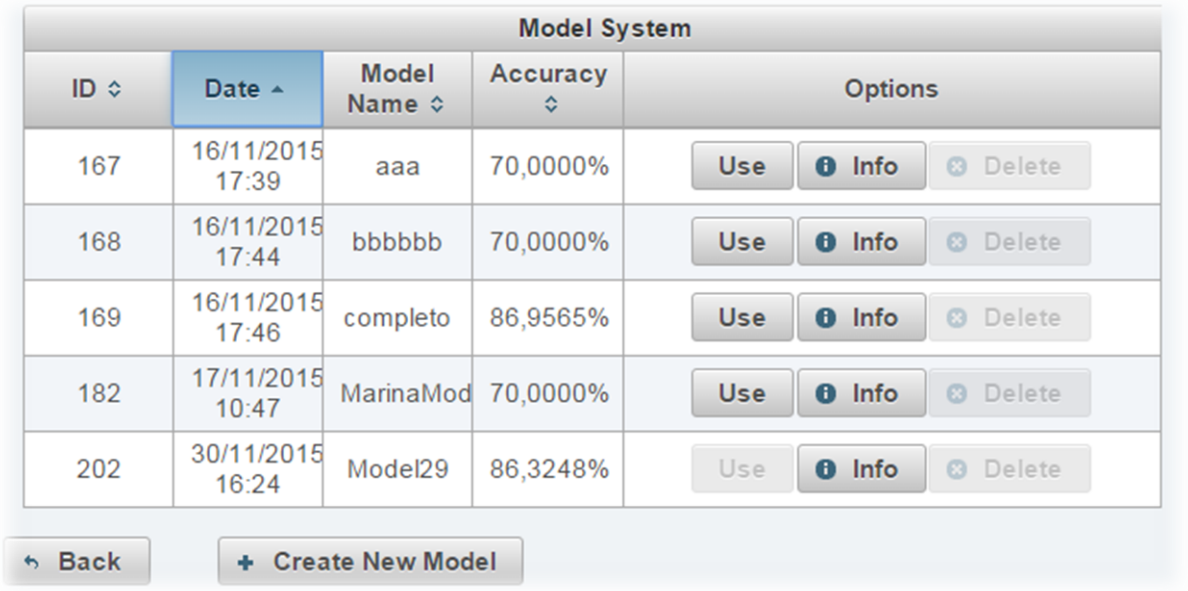
Una vez finalizado el estudio del tumor y sabiendo al 100% su diagnóstico, el médico procederá a asignar el diagnostico final a la citología. Gracias a esta asignación el sistema podrá utilizarla en la fase de creación de un modelo de clasificación.



Para ver los detalles de los parámetros de cada citología únicamente necesitamos pulsar sobre el icono Lupa.

9.3. Usuario Analytics Doctor

Si accedemos mediante el usuario Jorge con el rol Doctor Analyst, el sistema nos mostrará una lista de los modelos de clasificación existentes.

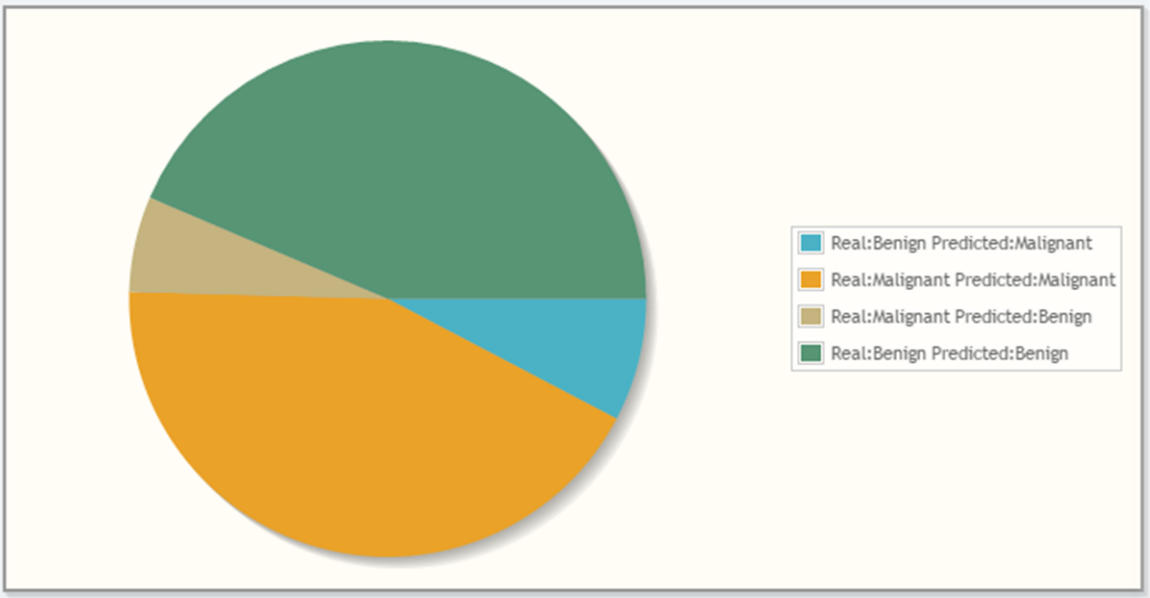


El modelo de clasificación que está usando actualmente el sistema es aquél donde el botón “Use” se encuentra deshabilitado. En el caso de que queramos seleccionar otro modelo para realizar las clasificaciones del sistema solo tenemos que pulsar sobre su botón “Use”.

También podemos ver las características de cada modelo pulsando sobre el botón “Info”. Al pulsar sobre este, el sistema nos muestra la siguiente información:

Confusion Matrix		
Real Diagnostic	Predicted Diagnostic	Number of Cases
Benign	Malignant	9
Malignant	Malignant	50
Malignant	Benign	7
Benign	Benign	51

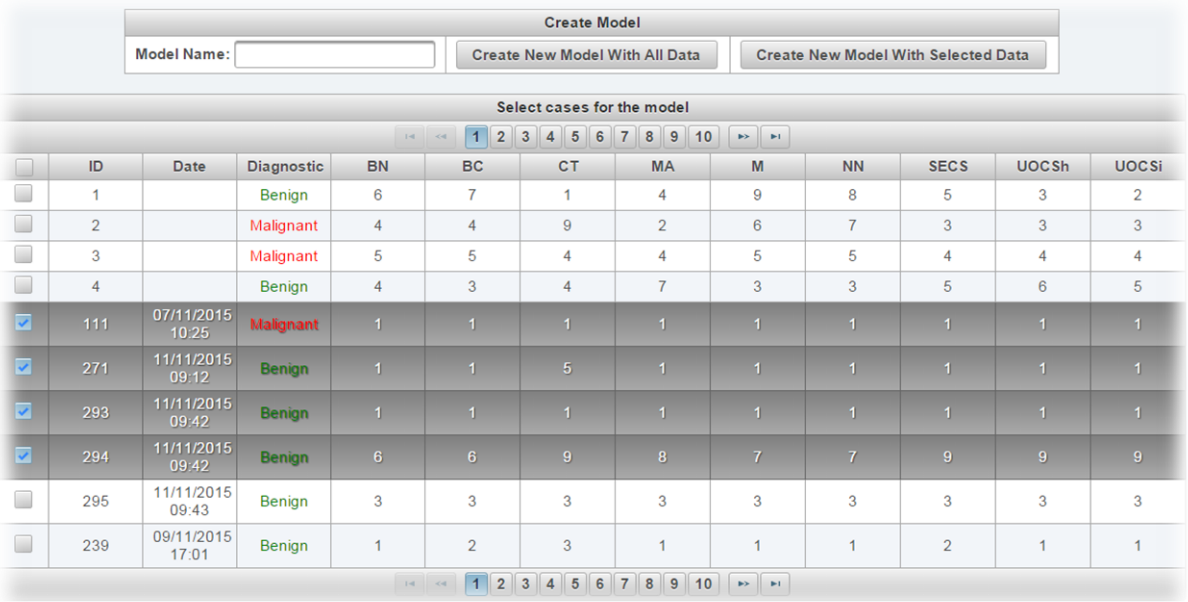
Chart



Back

En él podemos observar la matriz de confusión del modelo en una tabla y en forma de gráfica.

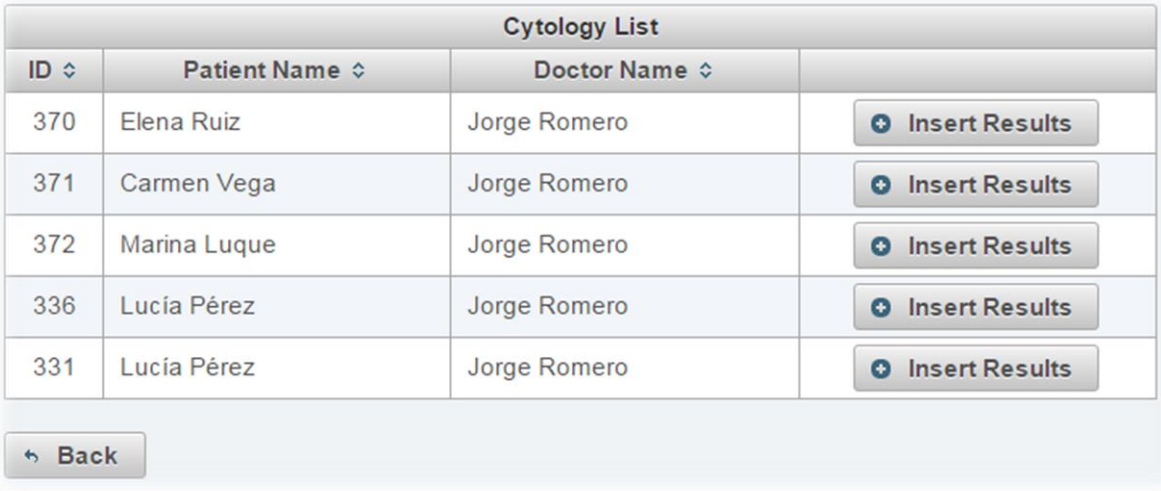
Si pulsamos sobre el botón “Create New Model” el sistema nos redirige a la siguiente vista. En ella podemos ver el listado completo de los resultados de todas las citologías del sistema. La creación de un modelo puede ser realizada de dos maneras. La primera es crear un modelo de clasificación a partir de todos los datos almacenados en el sistema. La segunda es crear el modelo a partir de los datos seleccionados manualmente.



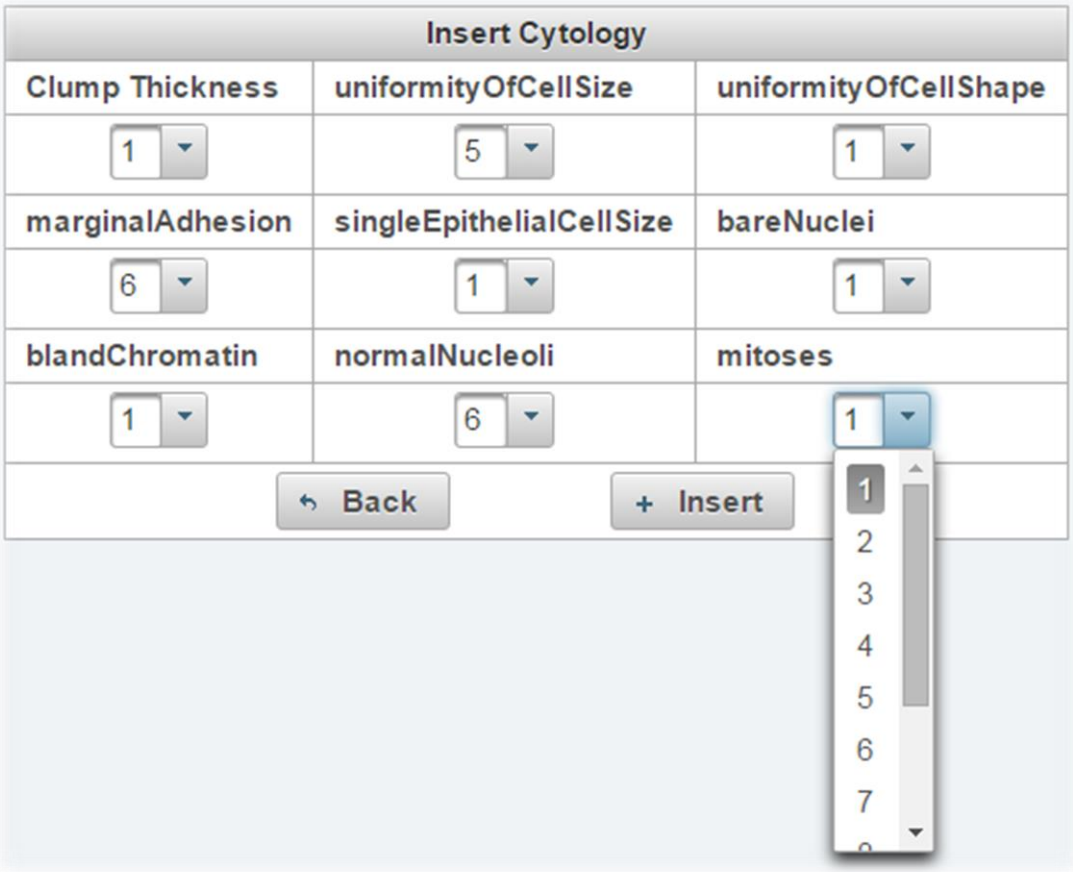
Una vez creado el modelo, el sistema nos redirige a la lista de modelos del sistema. En ella podremos ver las características de nuestro nuevo modelo y seleccionarlo como modelo de clasificación para el sistema.

9.4. Usuario Pathologist

Si accedemos al Sistema con el usuario Lidia y el rol Pathologist, el Sistema nos mostrará la lista de citologías pendientes que tiene la anatomopatóloga.



Una vez realizada la citología, la anatomopatóloga únicamente tiene que pulsar sobre el botón “Insert Results” del usuario pertinente e introducir los datos de la misma.



Una vez introducidos los datos, el sistema realiza automáticamente la predicción del diagnóstico y de la probabilidad de que este sea cierto.

Finalmente la citología terminada es enviada al médico pertinente.

10. Conclusiones

Las conclusiones obtenidas después del desarrollo del proyecto son las siguientes:

Oracle Data Miner ofrece una gran versatilidad sea cual sea el perfil del usuario que lo utilice. Por ejemplo, un investigador puede hacer uso de la herramienta gráfica sin la necesidad de tener conocimientos sobre SQL o PL/SQL. Él sólo necesitaría crear sus flujos de datos para la minería.

También existe el caso de ofrecerles a los usuarios menos avanzados o que tienen que hacer un uso básico de explotación datos, el flujo de trabajo ya diseñado para que solo introduzca los datos de entrada y analice los de salida.

Los usuarios avanzados pueden hacer uso de todas las herramientas combinadas. Como por ejemplo, las pruebas y creación de modelos desde la herramienta visual, ya que es la forma más simple y productiva. Y posteriormente utilizar estos modelos desde PL/SQL.

Finalmente, gracias a las librerías PL/SQL podemos realizar desarrollos de aplicaciones a nuestra medida (Como el desarrollado en este proyecto). Gracias a ello, se puede integrar la minería de datos en cualquier tipo de modelo de negocio y además adecuándola al contexto donde se utiliza (banca, medicina, investigación).

Por todo ello, desde mi punto de vista como alumno, considero que es una plataforma apta para la enseñanza de los procesos de minería de datos, gracias a sus múltiples niveles de uso y puntos de vista.

11. Bibliografía

11.1. Biografía principal

[1]	
Nombre	Predictive Analytics Using Oracle Data Miner
Autores	Brendan Tierney
Editorial	Oracle Press
Edición	2014

[2]	
Nombre	Building Data Mining Applications for CRM
Autores	Alex Berson, Stephen Smith, and Kurt Thearling
Editorial	MacGraw-Hill
Edición	1999

11.2. Bibliografía complementaría

[1]	
Nombre	Apuntes lenguaje PL/SQL
Autores	Manuel Enciso García-Oliveros

[2]	
Nombre	Apuntes asignatura de bases de datos
Autores	Manuel Enciso García-Oliveros

[3]	
Nombre	Apuntes asignatura modelado y diseño

[4]	
Nombre	Apuntes asignatura Ingeniería del software

[5]	
Nombre	Apuntes asignatura ingeniería de requisitos

11.3. Fuentes electrónicas

[1]	
Nombre	Minería de datos
Dirección	https://es.wikipedia.org/wiki/Miner%C3%ADa_de_datos

[2]	
Nombre	Repositorio Machine Learning de la Universidad de Irvine
Dirección	http://archive.ics.uci.edu/ml/

[3]	
Nombre	Oracle Data Miner
Dirección	http://www.oracle.com/technetwork/database/options/advancedanalytics/odm/dataminerworkflow-168677.html

[4]	
Nombre	Lenguaje Java
Dirección	https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)

[5]	
Nombre	Framework Java Server Faces
Dirección	https://es.wikipedia.org/wiki/JavaServer_Faces

[6]	
Nombre	Framework JPA
Dirección	https://es.wikipedia.org/wiki/Java_Persistence_API

[7]	
Nombre	Lenguaje HTML
Dirección	https://es.wikipedia.org/wiki/HTML

[8]	
Nombre	Lenguaje CSS
Dirección	https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada

[9]	
Nombre	Lenguaje SQL
Dirección	https://es.wikipedia.org/wiki/SQL

[10]	
Nombre	PL/SQL
Dirección	https://es.wikipedia.org/wiki/PL/SQL

[11]	
Nombre	Librería PrimeFaces
Dirección	https://es.wikipedia.org/wiki/PrimeFaces

[12]	
Nombre	Librería DBMS_DATA_MINING
Dirección	https://docs.oracle.com/cd/B13789_01/appdev.101/b10802/d_datmin.htm

[13]	
Nombre	Sqldeveloper
Dirección	http://www.oracle.com/technetwork/developer-tools/sqldeveloper/overview/index-097090.html

[14]	
Nombre	Datamodeler
Dirección	http://www.oracle.com/technetwork/developer-tools/datamodeler/overview/index.html

[15]	
Nombre	Netbeans
Dirección	https://es.wikipedia.org/wiki/NetBeans

[16]	
Nombre	MagicDraw
Dirección	https://en.wikipedia.org/wiki/MagicDraw

[17]	
Nombre	Google Chrome
Dirección	https://es.wikipedia.org/wiki/Google_Chrome

[18]	
Nombre	IBM Oracle Data Mining
Dirección	http://www-01.ibm.com/support/knowledgecenter/SS3RA7_17.0.0/clementine/dbmining_oracle_container.dita?lang=es

[19]	
Nombre	Tipos de problemas de minería de datos
Dirección	http://www.dataprix.com/tipos-de-problemas-de-mineria-de-datos

12. Anexo 1- Tecnologías utilizadas

12.1. Lenguajes

12.1.1. Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados.

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

La compañía Sun desarrolló la implementación de referencia original para los compiladores de Java, máquinas virtuales, y librerías de clases en 1991 y las publicó por primera vez en 1995. A partir de mayo de 2007, en cumplimiento con las especificaciones del Proceso de la Comunidad Java, Sun volvió a licenciar la mayoría de sus tecnologías de Java bajo la Licencia Pública General de GNU. Otros también han desarrollado implementaciones alternas a estas tecnologías de Sun, tales como el Compilador de Java de GNU y el GNU Classpath.

12.1.2. HTML

HTML, siglas de HyperText Markup Language («lenguaje de marcas de hipertexto»), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, entre otros. Es un estándar a cargo de la W3C, organización dedicada a la estandarización

de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web. Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

12.1.3. CSS

Hoja de estilo en cascada o CSS (siglas en inglés de cascading style sheets) es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML2 (y por extensión en XHTML). El World Wide Web Consortium (W3C) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

La información de estilo puede ser definida en un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales con el elemento «style» o en cada etiqueta particular mediante el atributo «style».

12.1.4. SQL

SQL (por sus siglas en inglés Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como hacer cambios en ellas.

12.1.5. PL/SQL

PL/SQL (Procedural Language/Structured Query Language) es un lenguaje de programación incrustado en Oracle.

PL/SQL soporta todas las consultas, ya que la manipulación de datos que se usa es la misma que en SQL, incluyendo nuevas características:

- El manejo de variables.
- Estructuras modulares.
- Estructuras de control de flujo y toma de decisiones.
- Control de excepciones.

El lenguaje PL/SQL está incorporado en:

- Servidor de la base de datos.

- Herramientas de Oracle (Forms, Reports,...).

En un entorno de base de datos los programadores pueden construir bloques PL/SQL para utilizarlos como procedimientos o funciones, o bien pueden escribir estos bloques como parte de scripts SQL*Plus.

Los programas o paquetes de PL/SQL se pueden almacenar en la base de datos como otro objeto, y todos los usuarios que estén autorizados tienen acceso a estos paquetes. Los programas se ejecutan en el servidor para ahorrar recursos a los clientes.

12.1.6. UML

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group).

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

12.2. Frameworks

12.2.1. JSF

JavaServer Faces (JSF) es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL (acrónimo de XML-based User-interface Language, lenguaje basado en XML para la interfaz de usuario)

JSF incluye:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
- Un conjunto por defecto de componentes para la interfaz de usuario.

- Dos bibliotecas de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Administración de estados.
- Beans administrados.

12.2.2. Java Persistence API

Java Persistence API, más conocida por sus siglas JPA, es la API de persistencia desarrollada para la plataformaJava EE

Es un framework del lenguaje de programación Java que maneja datos relacionales en aplicaciones usando la Plataforma Java en sus ediciones Standard (Java SE) y Enterprise (Java EE).

La JPA fue originada a partir del trabajo del JSR 220 Expert Group. Ha sido incluida en el estándar EJB3.

Persistencia en este contexto cubre tres áreas:

- La API en sí misma, definida en el paquete javax.persistence
- El lenguaje de consulta Java Persistence Query Language (JPQL).
- Metadatos objeto/relacional.

El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos (siguiendo el patrón de mapeo objeto-relacional), como sí pasaba con EJB2, y permitir usar objetos regulares (conocidos como POJOs).

12.3. Librerías

12.3.1. PrimeFaces

PrimeFaces es una librería de componentes para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Primefaces está bajo la licencia de Apache License V2.

Propiedades:

- Conjunto de componentes ricos (Editor de HTML, autocompletar, cartas, gráficas o paneles, entre otros).
- Soporte de ajax con despliegue parcial, lo que permite controlar qué componentes de la página actual se actualizarán y cuáles no.
- Componente para desarrollar aplicaciones web para teléfonos móviles, especiales para iPhones, Palm, Android y teléfonos móviles Nokia.

12.3.2. DBMS_DATA_MINING

Librería interna de Oracle que almacena todas las funciones y procedimientos necesarios para el uso la minería de datos en los desarrollos en PL/SQL.

Aquí una muestra de los procedimientos más importantes que contiene:

Procedimiento / Función	Propósito
APPLY Procedure	Aplicación del modelo
CREATE_MODEL Procedure	Creación de un modelo
COMPUTE_CONFUSION_MATRIX Procedure	Creación de la matriz de confusión
COMPUTE_LIFT Procedure	Calculo del LIFT
COMPUTE_ROC Procedure	Calculo de ROC
DROP_MODEL Procedure	Eliminación de un modelo
EXPORT_MODEL Procedure	Exportación de uno o más modelos del esquema
GET_ASSOCIATION_RULES Function	Esta función retorna las reglas de asociación de un modelo de asociación.
GET_DEFAULT_SETTINGS Function	Esta función retorna todas las opciones por defecto para todas las funciones y algoritmos de minería de datos.
GET_FREQUENT_ITEMSETS Function	Retorna un conjunto de filas que representan la frecuencia de un conjunto de datos perteneciente a un modelo de asociación
GET_MODEL_DETAILS_ABN Function	Proporciona los detalles de un modelo de Bayes adaptativo de red.
GET_MODEL_DETAILS_KM Function	Proporciona los detalles de un modelo K-Means
GET_MODEL_DETAILS_NB Function	Proporciona los detalles de un modelo Naive Bayes
GET_MODEL_DETAILS_NMF Function	Proporciona los detalles de un modelo NMF
GET_MODEL_DETAILS_SVM Function	Proporciona los detalles de un modelo SVM con kernel lineal

GET_MODEL_SETTINGS Function	Proporciona las opciones usadas para la creación de un modelo
GET_MODEL_SIGNATURE Function	Proporciona el identificador del modelo
IMPORT_MODEL Procedure	Importa uno o más modelos al esquema
RENAME_MODEL Procedure	Renombra un modelo

12.4. Herramientas

12.4.1. Oracle SQL Developer

Oracle SQL Developer es un entorno de desarrollo gratuito que simplifica el desarrollo y el mantenimiento de las bases de datos Oracle, ya sea de la manera tradicional o en la nube. Oracle SQL Developer ofrece un completo entorno de desarrollo para el lenguaje PL/SQL además de ser una solución completa para modelado datos y migración de estos de forma sencilla.

12.4.2. Oracle SQL Developer Data Modeler

Oracle SQL Developer Data Modeler es una aplicación gráfica gratuita para la realización de tareas de modelado. Usando SQL Develper Data Modeler los usuarios pueden crear, visualizar y editar, datos relacionales, lógicos y físicos. Además la aplicación proporciona herramientas de ingeniería inversa, facilitando el paso de unos modelos de datos a otros.

12.4.3. Netbeans

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE2 es un producto libre y gratuito sin restricciones de uso.

NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

12.4.4. MagicDraw

MagicDraw es una aplicación de modelado visual para UML, SysML, BPMN, y UPDM. Esta aplicación está diseñada para analistas de negocio, analistas de software, programadores e ingenieros de calidad. La aplicación simplifica el análisis y el diseño de sistemas orientados a objetos y bases de datos.

12.4.5. Google Chrome

Google Chrome es un navegador web desarrollado por Google y compilado con base en varios componentes e infraestructuras de desarrollo de aplicaciones (frameworks) de código abierto, como el motor de renderizado Blink (bifurcación o fork de WebKit).

13. Anexo 2 – Script de creación de la base de datos

En el siguiente script se proporciona la estructura de la base de datos, los procedimientos almacenados junto con la información necesaria para la correcta ejecución de la aplicación BreastCancerApp.

13.1. Estructura de la base de datos

```
--------------------------------------------------------
--  DDL for Sequence ID_CYTOLOGY
--------------------------------------------------------

   CREATE SEQUENCE  "DMUSER"."ID_CYTOLOGY"  MINVALUE 1 MAXVALUE
9999999999999999999999999999 INCREMENT BY 1 START WITH 410 CACHE 20 NOORDER
NOCYCLE ;
--------------------------------------------------------
--  DDL for Sequence ID_MODEL
--------------------------------------------------------

   CREATE SEQUENCE  "DMUSER"."ID_MODEL"  MINVALUE 1 MAXVALUE
9999999999999999999999999999 INCREMENT BY 1 START WITH 262 CACHE 20 NOORDER
NOCYCLE ;
--------------------------------------------------------
--  DDL for Sequence ID_MUEBLE
--------------------------------------------------------

   CREATE SEQUENCE  "DMUSER"."ID_MUEBLE"  MINVALUE 1 MAXVALUE
9999999999999999999999999999 INCREMENT BY 1 START WITH 30 CACHE 20 NOORDER
NOCYCLE ;
--------------------------------------------------------
--  DDL for Table MODEL
--------------------------------------------------------

  CREATE TABLE "DMUSER"."MODEL"
   (  "ID" NUMBER,
      "NAME" VARCHAR2(100 BYTE),
      "ACCURACY" FLOAT(126),
      "USING_" CHAR(1 BYTE) DEFAULT 0,
      "DATE_" DATE
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Table MODEL_TMP
--------------------------------------------------------

  CREATE TABLE "DMUSER"."MODEL_TMP"
   (  "ID" NUMBER,
      "CLUMP_THICKNESS" NUMBER,
      "UNIFORMITY_OF_CELL_SIZE" NUMBER,
      "UNIFORMITY_OF_CELL_SHAPE" NUMBER,
      "MARGINAL_ADHESION" NUMBER,
```

```
      "SINGLE_EPITHELIAL_CELL_SIZE" NUMBER,
      "BARE_NUCLEI" NUMBER,
      "BLAND_CHROMATIN" NUMBER,
      "NORMAL_NUCLEOLI" NUMBER,
      "MITOSES" NUMBER,
      "DIAGNOSTIC" NUMBER
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Table MODEL_TMP_SETTINGS
--------------------------------------------------------

  CREATE TABLE "DMUSER"."MODEL_TMP_SETTINGS"
   (  "SETTING_NAME" VARCHAR2(30 BYTE),
      "SETTING_VALUE" VARCHAR2(4000 BYTE)
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Table DOCTOR
--------------------------------------------------------

  CREATE TABLE "DMUSER"."DOCTOR"
   (  "ID" NUMBER,
      "NAME" VARCHAR2(100 BYTE),
      "SURNAME" VARCHAR2(100 BYTE),
      "PASSWORD" VARCHAR2(100 BYTE)
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Table PATIENT
--------------------------------------------------------

  CREATE TABLE "DMUSER"."PATIENT"
   (  "ID" NUMBER,
      "NAME" VARCHAR2(100 BYTE),
      "SURNAME" VARCHAR2(100 BYTE),
      "DOCTOR" NUMBER
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Table PATHOLOGIST
--------------------------------------------------------

  CREATE TABLE "DMUSER"."PATHOLOGIST"
   (  "ID" NUMBER,
```

```
      "NAME" VARCHAR2(100 BYTE),
      "SURNAME" VARCHAR2(100 BYTE),
      "PASSWORD" VARCHAR2(100 BYTE)
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Table CYTOLOGY
--------------------------------------------------------

  CREATE TABLE "DMUSER"."CYTOLOGY"
   (  "ID" NUMBER,
      "CLUMP_THICKNESS" NUMBER,
      "UNIFORMITY_OF_CELL_SIZE" NUMBER,
      "UNIFORMITY_OF_CELL_SHAPE" NUMBER,
      "MARGINAL_ADHESION" NUMBER,
      "SINGLE_EPITHELIAL_CELL_SIZE" NUMBER,
      "BARE_NUCLEI" NUMBER,
      "BLAND_CHROMATIN" NUMBER,
      "NORMAL_NUCLEOLI" NUMBER,
      "MITOSES" NUMBER,
      "DIAGNOSTIC" NUMBER,
      "PREDICTION_" NUMBER,
      "PROBABILITY_" FLOAT(126),
      "PATIENT" NUMBER,
      "PATHOLOGIST" NUMBER,
      "MODEL_" NUMBER,
      "DATE_" DATE,
      "NOTIFIED" CHAR(1 BYTE)
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;

--------------------------------------------------------
--  DDL for View MODEL_TMP_TEST_RESULTS
--------------------------------------------------------

  CREATE OR REPLACE FORCE VIEW "DMUSER"."MODEL_TMP_TEST_RESULTS" ("ID",
"DIAGNOSTIC", "PREDICTED_VALUE", "PROBABILITY") AS
  SELECT id,diagnostic,
        prediction(Model_20 USING * ) predicted_value,
        prediction_probability(Model_20 USING * ) probability
FROM model_tmp;
REM INSERTING into DMUSER.MODEL
SET DEFINE OFF;
```

13.2. Inserción de la información

```
Insert into DMUSER.MODEL (ID,NAME,ACCURACY,USING_,DATE_) values
('243','NB2','0,7','0',to_date('08/01/16','DD/MM/RR'));
Insert into DMUSER.MODEL (ID,NAME,ACCURACY,USING_,DATE_) values
('242','NB1','0,888888888888888888888888888888888889','1',to_date('08/01/
16','DD/MM/RR'));
Insert into DMUSER.MODEL (ID,NAME,ACCURACY,USING_,DATE_) values
('244','NB3','0,95','0',to_date('08/01/16','DD/MM/RR'));
Insert into DMUSER.MODEL (ID,NAME,ACCURACY,USING_,DATE_) values
('245','Model_20','1','0',to_date('08/01/16','DD/MM/RR'));
REM INSERTING into DMUSER.MODEL_TMP
SET DEFINE OFF;
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('139','5','7','7','1','5','8','3','4','1','2');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('140','10','5','8','10','3','10','5','1','3','4');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('141','5','10','10','6','10','10','10','6','5','4');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('142','8','8','9','4','5','10','7','8','1','4');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('143','10','4','4','10','6','10','5','5','1','4');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('144','7','9','4','10','10','3','5','3','3','4');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('145','5','1','4','1','2','1','3','2','1','2');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('146','10','10','6','3','3','10','4','3','2','4');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('147','3','3','5','2','3','10','7','1','1','4');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
```

```
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('148','10','8','8','2','3','4','8','7','8','4');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('179','9','1','2','6','4','10','7','7','2','4');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('180','8','4','10','5','4','4','7','10','1','4');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('181','1','1','1','1','2','1','3','1','1','2');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('182','10','10','10','7','9','10','7','10','10','4');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('183','1','1','1','1','2','1','3','1','1','2');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('184','8','3','4','9','3','10','3','3','1','4');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('185','10','8','4','4','4','10','3','10','4','4');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('186','1','1','1','1','2','1','3','1','1','2');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('187','1','1','1','1','2','1','3','1','1','2');
Insert into DMUSER.MODEL_TMP
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC) values
('188','7','8','7','6','4','3','8','8','4','4');
REM INSERTING into DMUSER.MODEL_TMP_SETTINGS
SET DEFINE OFF;
Insert into DMUSER.MODEL_TMP_SETTINGS (SETTING_NAME,SETTING_VALUE) values
('ALGO_NAME','ALGO_NAIVE_BAYES');
Insert into DMUSER.MODEL_TMP_SETTINGS (SETTING_NAME,SETTING_VALUE) values
('PREP_AUTO','ON');
REM INSERTING into DMUSER.DOCTOR
SET DEFINE OFF;
```

```
Insert into DMUSER.DOCTOR (ID,NAME,SURNAME,PASSWORD) values
('1','Jorge','Romero','1234');
REM INSERTING into DMUSER.PATIENT
SET DEFINE OFF;
Insert into DMUSER.PATIENT (ID,NAME,SURNAME,DOCTOR) values
('1','Elena','Ruiz','1');
Insert into DMUSER.PATIENT (ID,NAME,SURNAME,DOCTOR) values
('2','Carmen','Vega','1');
Insert into DMUSER.PATIENT (ID,NAME,SURNAME,DOCTOR) values
('3','Marina','Luque','1');
Insert into DMUSER.PATIENT (ID,NAME,SURNAME,DOCTOR) values
('4','Lucía','Pérez','1');
Insert into DMUSER.PATIENT (ID,NAME,SURNAME,DOCTOR) values
('34',null,null,null);
REM INSERTING into DMUSER.PATHOLOGIST
SET DEFINE OFF;
Insert into DMUSER.PATHOLOGIST (ID,NAME,SURNAME,PASSWORD) values
('1','Lidia','Aguilar','1234');
REM INSERTING into DMUSER.CYTOLOGY
SET DEFINE OFF;
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('139','5','7','7','1','5','8','3','4','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('140','10','5','8','10','3','10','5','1','3','4',null,null,null,null,null,
null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('141','5','10','10','6','10','10','10','6','5','4',null,null,null,null,nul
l,null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('142','8','8','9','4','5','10','7','8','1','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('143','10','4','4','10','6','10','5','5','1','4',null,null,null,null,null,
null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
```

```
('144','7','9','4','10','10','3','5','3','3','4',null,null,null,null,null,n
ull,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('145','5','1','4','1','2','1','3','2','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('146','10','10','6','3','3','10','4','3','2','4',null,null,null,null,null,
null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('147','3','3','5','2','3','10','7','1','1','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('148','10','8','8','2','3','4','8','7','8','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('149','1','1','1','1','2','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('150','8','4','7','1','3','10','3','9','2','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('151','5','1','1','1','2','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('152','3','3','5','2','3','10','7','1','1','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
```

```
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('153','7','2','4','1','3','4','3','3','1','4',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('154','3','1','1','1','2','1','3','2','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values ('155','3','1','3','1','2','-
1','2','1','1','2',null,null,null,null,null,null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('156','3','1','1','1','2','1','2','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('157','1','1','1','1','2','1','2','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('158','1','1','1','1','2','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('159','10','5','7','3','3','7','3','3','8','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('160','3','1','1','1','2','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('161','2','1','1','2','2','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
```

```
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('162','1','4','3','10','4','10','5','6','1','4',null,null,null,null,null,n
ull,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('163','10','4','6','1','2','10','5','3','1','4',null,null,null,null,null,n
ull,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('164','7','4','5','10','2','10','3','8','2','4',null,null,null,null,null,n
ull,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('165','8','10','10','10','8','10','10','7','3','4',null,null,null,null,nul
l,null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('166','10','10','10','10','10','10','4','10','10','4',null,null,null,null,
null,null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('167','3','1','1','1','3','1','2','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('168','6','1','3','1','4','5','5','10','1','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('169','5','6','6','8','6','10','4','10','4','4',null,null,null,null,null,n
ull,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('170','1','1','1','1','2','1','1','1','1','2',null,null,null,null,null,nul
l,null);
```

```
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('171','1','1','1','1','2','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values ('172','8','8','8','1','2','-
1','6','10','1','4',null,null,null,null,null,null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('173','10','4','4','6','2','10','2','3','1','4',null,null,null,null,null,n
ull,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values ('174','1','1','1','1','2','-
1','2','1','1','2',null,null,null,null,null,null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('175','5','5','7','8','6','10','7','4','1','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('176','5','3','4','3','4','5','4','7','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values ('177','5','4','3','1','2','-
1','2','3','1','2',null,null,null,null,null,null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('178','8','2','1','1','5','1','1','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('179','9','1','2','6','4','10','7','7','2','4',null,null,null,null,null,nu
ll,null);
```

```
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('180','8','4','10','5','4','4','7','10','1','4',null,null,null,null,null,n
ull,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('181','1','1','1','1','2','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('182','10','10','10','7','9','10','7','10','10','4',null,null,null,null,nu
ll,null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('183','1','1','1','1','2','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('184','8','3','4','9','3','10','3','3','1','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('185','10','8','4','4','4','10','3','10','4','4',null,null,null,null,null,
null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('186','1','1','1','1','2','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('187','1','1','1','1','2','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
```

```
('188','7','8','7','6','4','3','8','8','4','4',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('189','3','1','1','1','2','5','5','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('190','2','1','1','1','3','1','2','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('191','1','1','1','1','2','1','1','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('192','8','6','4','10','10','1','3','5','1','4',null,null,null,null,null,n
ull,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('193','1','1','1','1','2','1','1','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('194','1','1','1','1','1','1','2','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values ('195','4','6','5','6','7','-
1','4','9','1','2',null,null,null,null,null,null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('196','5','5','5','2','5','10','4','3','1','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
```

```
DEL_,DATE_,NOTIFIED) values
('197','6','8','7','8','6','8','8','9','1','4',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('198','1','1','1','1','5','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('199','4','4','4','4','6','5','7','3','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('200','7','6','3','2','5','10','7','4','6','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values ('201','3','1','1','1','2','-
1','3','1','1','2',null,null,null,null,null,null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('202','3','1','1','1','2','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('203','5','4','6','10','2','10','4','1','1','4',null,null,null,null,null,n
ull,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('204','1','1','1','1','2','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('205','3','2','2','1','2','1','2','3','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
```

```
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('206','10','1','1','1','2','10','5','4','1','4',null,null,null,null,null,n
ull,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('207','1','1','1','1','2','1','2','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('208','8','10','3','2','6','4','3','10','1','4',null,null,null,null,null,n
ull,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('209','10','4','6','4','5','10','7','1','1','4',null,null,null,null,null,n
ull,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('210','10','4','7','2','2','8','6','1','1','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('211','5','1','1','1','2','1','3','1','2','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('212','5','2','2','2','2','1','2','2','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('213','5','4','6','6','4','10','4','3','1','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('214','8','6','7','3','3','10','3','4','2','4',null,null,null,null,null,nu
ll,null);
```

```
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('215','1','1','1','1','2','1','1','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('216','6','5','5','8','4','10','3','4','1','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('217','1','1','1','1','2','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('218','1','1','1','1','1','1','2','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('219','8','5','5','5','2','10','4','3','1','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('220','10','3','3','1','2','10','7','6','1','4',null,null,null,null,null,n
ull,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('221','1','1','1','1','2','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('222','2','1','1','1','2','1','1','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
```

```
('223','1','1','1','1','2','1','1','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('224','7','6','4','8','10','10','9','5','3','4',null,null,null,null,null,n
ull,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('225','1','1','1','1','2','1','1','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('226','5','2','2','2','3','1','1','3','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('227','1','1','1','1','1','1','1','3','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('228','3','4','4','10','5','1','3','3','1','4',null,null,null,null,null,nu
ll,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('229','4','2','3','5','3','8','7','6','1','4',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('230','5','1','1','3','2','1','1','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('231','2','1','1','1','2','1','3','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
```

```
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('232','3','4','5','3','7','3','4','6','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('233','2','7','10','10','7','10','4','9','4','4',null,null,null,null,null,
null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('234','1','1','1','1','2','1','2','1','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('235','4','1','1','1','3','1','2','2','1','2',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('236','5','3','3','1','3','3','3','3','3','4',null,null,null,null,null,nul
l,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('237','8','10','10','7','10','10','7','3','8','4',null,null,null,null,null
,null,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('238','8','10','5','3','8','4','4','10','3','4',null,null,null,null,null,n
ull,null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('393','6','5','6','4','1','5','6','7','4',null,'4','0,99998193979263306','
1','1','242',to_date('08/01/16','DD/MM/RR'),null);
Insert into DMUSER.CYTOLOGY
(ID,CLUMP_THICKNESS,UNIFORMITY_OF_CELL_SIZE,UNIFORMITY_OF_CELL_SHAPE,MARGIN
AL_ADHESION,SINGLE_EPITHELIAL_CELL_SIZE,BARE_NUCLEI,BLAND_CHROMATIN,NORMAL_
NUCLEOLI,MITOSES,DIAGNOSTIC,PREDICTION_,PROBABILITY_,PATIENT,PATHOLOGIST,MO
DEL_,DATE_,NOTIFIED) values
('392','6','5','6','4','1','5','6','7','4','4','4','0,99999994039535522','1
','1','244',to_date('08/01/16','DD/MM/RR'),null);
REM INSERTING into DMUSER.MODEL_TMP_TEST_RESULTS
```

```
SET DEFINE OFF;
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values
('139','2','2','0,99997133016586304');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('140','4','4','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('141','4','4','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('142','4','4','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('143','4','4','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('144','4','4','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('145','2','2','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('146','4','4','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('147','4','4','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('148','4','4','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('179','4','4','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('180','4','4','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('181','2','2','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('182','4','4','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('183','2','2','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('184','4','4','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('185','4','4','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('186','2','2','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('187','2','2','1');
Insert into DMUSER.MODEL_TMP_TEST_RESULTS
(ID,DIAGNOSTIC,PREDICTED_VALUE,PROBABILITY) values ('188','4','4','1');
```

13.3. Creación de Índices y Triggers

```
--------------------------------------------------------
--  DDL for Index MODEL_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "DMUSER"."MODEL_PK" ON "DMUSER"."MODEL" ("ID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index MODEL_UK1
--------------------------------------------------------

  CREATE UNIQUE INDEX "DMUSER"."MODEL_UK1" ON "DMUSER"."MODEL" ("NAME")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index MODEL_TMP_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "DMUSER"."MODEL_TMP_PK" ON "DMUSER"."MODEL_TMP"
("ID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index DOCTOR_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "DMUSER"."DOCTOR_PK" ON "DMUSER"."DOCTOR" ("ID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index PATIENT_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "DMUSER"."PATIENT_PK" ON "DMUSER"."PATIENT" ("ID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index PATHOLOGIST_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "DMUSER"."PATHOLOGIST_PK" ON "DMUSER"."PATHOLOGIST"
("ID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
```

```
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index CYTOLOGY_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "DMUSER"."CYTOLOGY_PK" ON "DMUSER"."CYTOLOGY" ("ID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  Constraints for Table MODEL
--------------------------------------------------------

  ALTER TABLE "DMUSER"."MODEL" ADD CONSTRAINT "MODEL_PK" PRIMARY KEY ("ID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;

  ALTER TABLE "DMUSER"."MODEL" ADD CONSTRAINT "MODEL_UK1" UNIQUE ("NAME")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;

  ALTER TABLE "DMUSER"."MODEL" MODIFY ("ID" NOT NULL ENABLE);
--------------------------------------------------------
--  Constraints for Table MODEL_TMP
--------------------------------------------------------

  ALTER TABLE "DMUSER"."MODEL_TMP" ADD CONSTRAINT "MODEL_TMP_PK" PRIMARY
KEY ("ID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;

  ALTER TABLE "DMUSER"."MODEL_TMP" MODIFY ("ID" NOT NULL ENABLE);
--------------------------------------------------------
--  Constraints for Table DOCTOR
--------------------------------------------------------

  ALTER TABLE "DMUSER"."DOCTOR" ADD CONSTRAINT "DOCTOR_PK" PRIMARY KEY
("ID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;

  ALTER TABLE "DMUSER"."DOCTOR" MODIFY ("ID" NOT NULL ENABLE);
--------------------------------------------------------
--  Constraints for Table PATIENT
```

```
--------------------------------------------------------

  ALTER TABLE "DMUSER"."PATIENT" ADD CONSTRAINT "PATIENT_PK" PRIMARY KEY
("ID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;

  ALTER TABLE "DMUSER"."PATIENT" MODIFY ("ID" NOT NULL ENABLE);
--------------------------------------------------------
--  Constraints for Table PATHOLOGIST
--------------------------------------------------------

  ALTER TABLE "DMUSER"."PATHOLOGIST" ADD CONSTRAINT "PATHOLOGIST_PK"
PRIMARY KEY ("ID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;

  ALTER TABLE "DMUSER"."PATHOLOGIST" MODIFY ("ID" NOT NULL ENABLE);
--------------------------------------------------------
--  Constraints for Table CYTOLOGY
--------------------------------------------------------

  ALTER TABLE "DMUSER"."CYTOLOGY" ADD CONSTRAINT "CYTOLOGY_PK" PRIMARY KEY
("ID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;

  ALTER TABLE "DMUSER"."CYTOLOGY" MODIFY ("ID" NOT NULL ENABLE);
--------------------------------------------------------
--  Ref Constraints for Table PATIENT
--------------------------------------------------------

  ALTER TABLE "DMUSER"."PATIENT" ADD CONSTRAINT "PATIENT_DOCTOR_FK" FOREIGN
KEY ("DOCTOR")
        REFERENCES "DMUSER"."DOCTOR" ("ID") ENABLE;
--------------------------------------------------------
--  Ref Constraints for Table CYTOLOGY
--------------------------------------------------------

  ALTER TABLE "DMUSER"."CYTOLOGY" ADD CONSTRAINT "CYTOLOGY_MODEL_FK"
FOREIGN KEY ("MODEL_")
        REFERENCES "DMUSER"."MODEL" ("ID") ENABLE;

  ALTER TABLE "DMUSER"."CYTOLOGY" ADD CONSTRAINT "CYTOLOGY_PATHOLOGIST_FK"
FOREIGN KEY ("PATHOLOGIST")
        REFERENCES "DMUSER"."PATHOLOGIST" ("ID") ENABLE;

  ALTER TABLE "DMUSER"."CYTOLOGY" ADD CONSTRAINT "CYTOLOGY_PATIENT_FK"
FOREIGN KEY ("PATIENT")
        REFERENCES "DMUSER"."PATIENT" ("ID") ENABLE;
--------------------------------------------------------
--  DDL for Trigger TRIG_ID_MODEL
```

```
--------------------------------------------------------

  CREATE OR REPLACE TRIGGER "DMUSER"."TRIG_ID_MODEL"
before insert on model
for each row
begin
select id_model.nextval into :new.id from dual;
END
;
/
ALTER TRIGGER "DMUSER"."TRIG_ID_MODEL" ENABLE;
--------------------------------------------------------
--  DDL for Trigger TRIG_ID_CYTOLOGY
--------------------------------------------------------

  CREATE OR REPLACE TRIGGER "DMUSER"."TRIG_ID_CYTOLOGY"
before insert on cytology
for each row
begin
select id_cytology.nextval into :new.id from dual;
END
;
/
ALTER TRIGGER "DMUSER"."TRIG_ID_CYTOLOGY" ENABLE;
```

13.4. Creación del procedimiento almacenado

El siguiente procedimiento almacenado es llamado cada vez que la aplicación BreastCancerApp realiza la predicción del diagnóstico de una paciente.

```
--------------------------------------------------------
--  DDL for Procedure CREATEMODEL
--------------------------------------------------------
set define off;

  CREATE OR REPLACE PROCEDURE "DMUSER"."CREATEMODEL" ( model_name in
varchar )  as
v_accuracy NUMBER;
begin

   SYS.DBMS_DATA_MINING.CREATE_MODEL(
    model_name => model_name,
    mining_function => dbms_data_mining.classification,
    data_table_name => 'MODEL_TMP',
    case_id_column_name => 'ID',
    target_column_name => 'DIAGNOSTIC',
    settings_table_name => 'model_tmp_settings'
  );

  execute immediate 'CREATE OR REPLACE VIEW model_tmp_test_results
AS
SELECT id,diagnostic,
        prediction('||model_name||' USING * ) predicted_value,
        prediction_probability('||model_name||' USING * ) probability
FROM model_tmp';
```

```
  DBMS_DATA_MINING.COMPUTE_CONFUSION_MATRIX(
    accuracy => v_accuracy,
    apply_result_table_name => 'model_tmp_test_results',
    target_table_name => 'model_tmp_test_results',
    case_id_column_name => 'id',
    target_column_name => 'DIAGNOSTIC',
    confusion_matrix_table_name => model_name||'_confusion_matrix',
    score_column_name => 'PREDICTED_VALUE',
    score_criterion_column_name => 'PROBABILITY',
    cost_matrix_table_name => null,
    apply_result_schema_name => null,
    target_schema_name => null,
    cost_matrix_schema_name => null,
    score_criterion_type => 'PROBABILITY'
  );

  insert into model (name, accuracy, date_) values (model_name,v_accuracy,
systimestamp);

end createmodel;
```

Agradecimientos

Agradecer en primer lugar a mis padres el haber estado siempre ahí y haberme apoyado siempre en la elección de mi vida profesional.

Agradecer a mi pareja, Lidia Aguilar Reyes, el haberme apoyado durante tanto tiempo y el haberme motivado siempre que lo necesitaba. Además de la gran ayuda que me ha ofrecido en la realización de este proyecto.

Agradecer a mi compañero David Doña Corrales el haber compartido esta gran aventura que es la informática. Quién nos hubiera dicho que llegaríamos aquí hace 7 años, mientras mirábamos con desolación aquel ofuscado código en C, casi a las 3 de la tarde. Cuando un puntero era nuestro mayor miedo. Cuando en un examen entraba la serie Fibonacci en forma recursiva y nos tirábamos de los pelos...

*Finalmente agradecer a todos los profesores la gran imaginación de la que hacen uso en sus exámenes, sin vosotros no hubiera sido lo mismo. Gracias a ello ya tengo muchas historias que contar a mis descendientes. Como cuando no podía coger el coche después de un examen de cálculo para no poner en peligro la seguridad de los viandantes. La DGT debería de hacer una campaña de concienciación con este tema. Da miedo ver a los alumnos con las manos en volante y la mirada perdida, pesando de dónde ** se había sacado el profesor aquella integral.*