Universidad Mariano Gálvez de Guatemala
Centro Universitario Retalhuleu
Ingeniería En Sistemas de la Información y Ciencias de la Computación
Base de Datos II
Ing. Jorge Santos

Minería de Datos - Algoritmos

Estudiante: Cesar Augusto Tuch Pacajoj

Carné: 2890-14-17950

Introducción:

En la actualidad las pequeñas, medianas y grandes empresas se basan a números para proyectar y predecir las ventas, los gastos, la fidelización de los clientes para la toma de decisiones es por ello que se necesita de la informática para eficientizar este tipo de procesos que si se hiciera manual tomaría mucho tiempo por lo que se necesita de la MINERIA DE DATOS y sus algoritmos para mejorar dichos procesos. La Minería de datos implementa nueve algoritmos de los cual se puede elegir alguno para el análisis de cantidades grandes de datos y dependiendo del resultado que se espere se puede elegir las tablas semejantes para que el análisis proporcione un mejor resultado.

Las empresas actualmente necesitan de información oportuna, veraz y confiable para la toma de decisiones, esta es necesaria para la proyección de ventas de un año, semestre, trimestre, bimestre o mensual por lo que la Minería de Datos se acopla perfectamente para este tipo de trabajo, en el siguiente desarrollo del tema se explica cada uno de los algoritmos con ejemplos y como se puede implementar, se aclara que deberá adaptarse a cada empresa y el analista es el que se encargara de elegir que tablas o parte de la empresa quiere analizar.

Minería de Datos:

Un *algoritmo* en minería de datos (o aprendizaje automático) es un conjunto de heurísticas y cálculos que permiten crear un modelo a partir de datos. Para crear un modelo, el algoritmo analiza primero los datos proporcionados, en busca de tipos específicos de patrones o tendencias. El algoritmo usa los resultados de este análisis en un gran número de iteraciones para determinar los parámetros óptimos para crear el modelo de minería de datos. A continuación, estos parámetros se aplican en todo el conjunto de datos para extraer patrones procesables y estadísticas detalladas.
El modelo de minería de datos que crea un algoritmo a partir de los datos puede tomar diversas formas, incluyendo:

- Un conjunto de clústeres que describe cómo se relacionan los casos de un conjunto de datos.
- Un árbol de decisión que predice un resultado y que describe cómo afectan a este los distintos criterios.
- Un modelo matemático que predice las ventas.
- Un conjunto de reglas que describen cómo se agrupan los productos en una transacción, y las probabilidades de que dichos productos se adquieran juntos.

Los algoritmos proporcionados en la minería de datos de SQL Server son los métodos más comunes y probados para derivar patrones a partir de datos. Por ejemplo, la agrupación en clústeres mediana-K es uno de los algoritmos de agrupación en clústeres más antiguo y está disponible en un gran número de herramientas y con diferentes implementaciones y opciones. Pero la implementación específica de la agrupación en clústeres mediana-K usada en la minería de datos de SQL Server ha sido desarrollada por Microsoft Research y se ha optimizado para rendimiento con Analysis Services. Todos los algoritmos de minería de datos de Microsoft se pueden personalizar ampliamente y usar mediante programación con las API proporcionadas. También puede automatizar la creación, aprendizaje y reciclaje de modelos con los componentes de minería de datos de Integration Services. Además, puede usar algoritmos de minería de datos desarrollados por terceros que cumplan con la especificación OLE DB para minería de datos, o bien desarrollar algoritmos personalizados que se puedan registrar como servicios para usarlos después en el marco de la minería de datos de SQL Server.

ALGORITMOS:

Algoritmo de Asociación de Microsoft:

El algoritmo de asociación de Microsoft es un algoritmo que suele usarse para los motores de recomendación. Un motor de recomendación recomienda elementos a los clientes basándose en los elementos que ya han adquirido o en los que tienen interés. El algoritmo de asociación de Microsoft también resulta útil para el análisis de la cesta de compra.
Los modelos de asociación se generan basándose en conjuntos de datos que contienen identificadores para casos individuales y para los elementos que contienen los casos. Un grupo de elementos de un caso se denomina un *conjunto de elementos*. Un modelo de asociación se compone de una serie de conjuntos de elementos y de las reglas que describen cómo estos elementos se agrupan dentro de los casos. Las reglas que el algoritmo identifica pueden utilizarse para predecir las probables compras de un cliente en el futuro, basándose en los elementos existentes en la cesta de

compra actual del cliente. El siguiente diagrama muestra una serie de reglas en un conjunto de elementos

Ejemplo

La empresa Adventure Works Cycle está rediseñando la funcionalidad de su sitio web. El objetivo del nuevo diseño es incrementar la venta directa de sus productos. Debido a que la empresa registra cada venta en una base de datos transaccional, se puede utilizar el algoritmo de asociación de Microsoft para identificar los conjuntos de productos que suelen adquirirse juntos. Así, se pueden predecir los elementos adicionales en los que un cliente puede estar interesado basándose en los elementos que ya se encuentran en su cesta de la compra.

Cómo Funciona el Algoritmo:

El algoritmo de asociación de Microsoft recorre un conjunto de datos para hallar elementos que aparezcan juntos en un caso. Después, agrupa en conjuntos de elementos todos los elementos asociados que aparecen, como mínimo, en el número de casos especificado en el parámetro *MINIMUM_SUPPORT*. Por ejemplo, un conjunto de elementos puede ser "Mountain 200=Existing, Sport 100=Existing" y tener un soporte de 710. El algoritmo generará reglas a partir de los conjuntos de elementos. Estas reglas se usan para predecir la presencia de un elemento en la base de datos, basándose en la presencia de otros elementos específicos que el algoritmo ha identificado como importantes. Por ejemplo, una regla puede ser "if Touring 1000=existing and Road bottle cage=existing, then Water bottle=existing", y puede tener una probabilidad de 0.812. En este ejemplo, el algoritmo identifica que la presencia en la cesta del neumático Touring 1000 y del soporte de la botella de agua predice que probablemente la cesta de compra incluirá también una botella de agua.

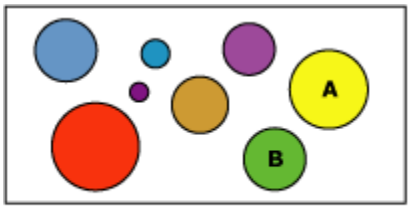
Datos requeridos para los modelos de asociación

Al preparar los datos para su uso en un modelo de reglas de asociación, conviene comprender qué requisitos son imprescindibles para el algoritmo concreto, incluidos el volumen de datos necesario y la forma en que estos datos se utilizan.
Los requisitos para un modelo de reglas de asociación son los siguientes:

- **Una columna de una sola clave** : cada modelo debe contener una columna numérica o de texto que identifique cada registro de manera única. no se permiten las claves compuestas.
- **Una única columna de predicción** Un modelo de asociación solo puede tener una columna de predicción. Normalmente, se trata de la columna de clave de la tabla anidada, como el campo que contiene los productos que se han comprado. Los valores deben ser discretos o discretizados.
- **Columnas de entrada** Las columnas de entrada deben ser discretas. Los datos de entrada de un modelo de asociación suelen encontrarse en dos tablas. Por ejemplo, una tabla puede contener la información del cliente y la otra las compras de ese cliente. Es posible incluir estos datos en el modelo mediante el uso de una tabla anidada. Para obtener más información sobre las tablas anidadas,

Algoritmo de Clústeres de Microsoft :

El algoritmo de clústeres de Microsoft es un algoritmo de *segmentación* o *clústeres* que itera en los casos de un conjunto de datos para agruparlos en clústeres que contengan características similares. Estas agrupaciones son útiles para la exploración de datos, la identificación de anomalías en los datos y la creación de predicciones.
Los modelos de agrupación en clústeres identifican las relaciones en un conjunto de datos que no se podrían derivar lógicamente a través de la observación casual. Por ejemplo, puede adivinar fácilmente que las personas que se desplazan a sus trabajos en bicicleta no viven, por lo general, a gran distancia de sus lugares de trabajo. Sin embargo, el algoritmo puede encontrar otras características que no son evidentes acerca de los trabajadores que se desplazan en bicicleta. En el siguiente diagrama, el clúster A representa los datos sobre las personas que suelen conducir hasta el trabajo, en tanto que el clúster B representa los datos sobre las personas que van hasta allí en bicicleta.



A = Trabajadores que conducen para ir al trabajo
B = Trabajadores que van en bicicleta al trabajo

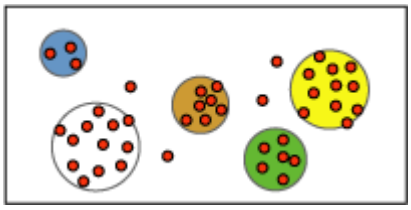
El algoritmo de clústeres se diferencia de otros algoritmos de minería de datos, como el algoritmo de árboles de decisión de Microsoft , en que no se tiene que designar una columna de predicción para generar un modelo de agrupación en clústeres. El algoritmo de clústeres entrena el modelo de forma estricta a partir de las relaciones que existen en los datos y de los clústeres que identifica el algoritmo.

Ejemplo

Considere un grupo de personas que comparten información demográfica similar y que adquieren productos similares de la empresa Adventure Works . Este grupo de personas representa un clúster de datos. En una base de datos pueden existir varios clústeres como éstos. Mediante la observación de las columnas que forman un clúster, puede ver con mayor claridad la forma en que los registros de un conjunto de datos se relacionan entre sí.

Cómo Funciona el Algoritmo:

El algoritmo de clústeres de Microsoft identifica primero las relaciones de un conjunto de datos y genera una serie de clústeres basándose en ellas. Un gráfico de dispersión es una forma útil de representar visualmente el modo en que el algoritmo agrupa los datos, tal como se muestra en el siguiente diagrama. El gráfico de dispersión representa todos los casos del conjunto de datos; cada caso es un punto del gráfico. Los clústeres agrupan los puntos del gráfico e ilustran las relaciones que identifica el algoritmo.



Después de definir los clústeres, el algoritmo calcula el grado de perfección con que los clústeres representan las agrupaciones de puntos y, a continuación, intenta volver a definir las agrupaciones para crear clústeres que representen mejor los datos. El algoritmo establece una iteración en este proceso hasta que ya no es posible mejorar los resultados mediante la redefinición de los clústeres.
Puede personalizar el funcionamiento del algoritmo seleccionando una técnica de agrupación en clústeres, limitando el número máximo de clústeres o cambiando la cantidad de soporte que se requiere para crear un clúster. Para obtener más información. Este algoritmo incluye dos métodos populares de agrupación en clústeres: el método de agrupación en clústeres k-means y el método de maximización de la expectativa.

Datos requeridos para los modelos de agrupación de Clusters:

Al preparar los datos para su uso en el entrenamiento de un modelo de agrupación en clústeres, conviene comprender qué requisitos son imprescindibles para el algoritmo concreto, incluidos el volumen de datos necesario y la forma en que estos datos se utilizan.
Los requisitos para un modelo de agrupación en clústeres son los siguientes:

- **Una columna de una sola clave** : cada modelo debe contener una columna numérica o de texto que identifique cada registro de manera única. No están permitidas las claves compuestas.
- **Columnas de entrada** Cada modelo debe tener al menos una columna de entrada que contenga los valores que se utilizan para generar los clústeres. Puede tener tantas columnas de entrada como desee, pero dependiendo del número de valores existentes en cada columna, la adición de columnas adicionales podría aumentar el tiempo necesario para entrenar el modelo.
- **Una columna de predicción opcional** El algoritmo no necesita una columna de predicción para generar el modelo, pero puede agregar una columna de predicción de casi cualquier tipo de datos. Los valores de la columna de predicción se pueden tratar como entradas del modelo de agrupación en clústeres, o se puede especificar que solo se utilicen para las

predicciones. Por ejemplo, si desea predecir los ingresos del cliente agrupando en clústeres de acuerdo con datos demográficos como la región o la edad, se deben especificar los ingresos como **PredictOnly** y agregar todas las demás columnas, como la región o la edad, como entradas.

Algoritmo de Arboles de Decisión:

El algoritmo de árboles de decisión de Microsoft es un algoritmo de clasificación y regresión para el modelado de predicción de atributos discretos y continuos.
Para los atributos discretos, el algoritmo hace predicciones basándose en las relaciones entre las columnas de entrada de un conjunto de datos. Utiliza los valores, conocidos como estados, de estas columnas para predecir los estados de una columna que se designa como elemento de predicción. Específicamente, el algoritmo identifica las columnas de entrada que se correlacionan con la columna de predicción. Por ejemplo, en un escenario para predecir qué clientes van a adquirir probablemente una bicicleta, si nueve de diez clientes jóvenes compran una bicicleta, pero solo lo hacen dos de diez clientes de edad mayor, el algoritmo infiere que la edad es un buen elemento de predicción en la compra de bicicletas. El árbol de decisión realiza predicciones basándose en la tendencia hacia un resultado concreto.
Para los atributos continuos, el algoritmo usa la regresión lineal para determinar dónde se divide un árbol de decisión.
Si se define más de una columna como elemento de predicción, o si los datos de entrada contienen una tabla anidada que se haya establecido como elemento de predicción, el algoritmo genera un árbol de decisión independiente para cada columna de predicción.

Ejemplo

El departamento de marketing de la empresa Adventure Works Cycles desea identificar las características de los clientes antiguos que podrían indicar si es probable que realicen alguna compra en el futuro. La base de datos AdventureWorks2012 almacena información demográfica que describe a los clientes antiguos. Mediante el algoritmo de árboles de decisión de Microsoft que analiza esta información, el departamento puede generar un modelo que predice si un determinado cliente va a comprar productos, basándose en el estado de las columnas conocidas sobre ese cliente, como la demografía o los patrones de compra anteriores.

Cómo Funciona el Algoritmo:

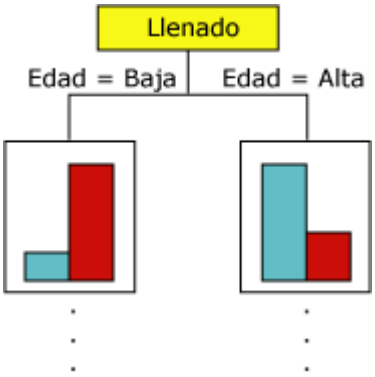
El algoritmo de árboles de decisión de Microsoft genera un modelo de minería de datos mediante la creación de una serie de divisiones en el árbol. Estas divisiones se representan como *nodos*. El algoritmo agrega un nodo al modelo cada vez que una columna de entrada tiene una correlación significativa con la columna de predicción. La forma en que el algoritmo determina una división varía en función de si predice una columna continua o una columna discreta.
El algoritmo de árboles de decisión de Microsoft utiliza la *selección de características* para guiar la selección de los atributos más útiles. Todos los algoritmos de minería de datos de SQL Server Data

Mining algorithms to improve performance and the quality of analysis. La selección de características es importante para evitar que los atributos irrelevantes utilicen tiempo de procesador. Si utiliza demasiados atributos de predicción o de entrada al diseñar un modelo de minería de datos, el modelo puede tardar mucho tiempo en procesarse o incluso quedarse sin memoria. Algunos métodos para determinar si hay que dividir el árbol son las métricas estándar del sector para la *entropía* y las redes Bayesianas. Para obtener más información sobre los métodos que se usan para seleccionar los atributos significativos y, después, puntuarlos y clasificarlos.
Un problema común de los modelos de minería de datos es que el modelo se vuelve demasiado sensible a las diferencias pequeñas en los datos de entrenamiento, en cuyo caso se dice que está *sobreajustado* o *sobreentrenado*. Un modelo sobreajustado no se puede generalizar a otros conjuntos de datos. Para evitar sobreajustar un conjunto de datos determinado, el algoritmo de árboles de decisión de Microsoft utiliza técnicas para controlar el crecimiento del árbol.

Predecir columnas discretas

La forma en que el algoritmo de árboles de decisión de Microsoft genera un árbol para una columna de predicción discreta puede mostrarse mediante un histograma. El siguiente diagrama muestra un histograma que seguimiento una columna de predicción, Bike Buyers, con una columna de entrada, Age. El histograma muestra que la edad de una persona ayuda a distinguir si esa persona comprará una bicicleta.

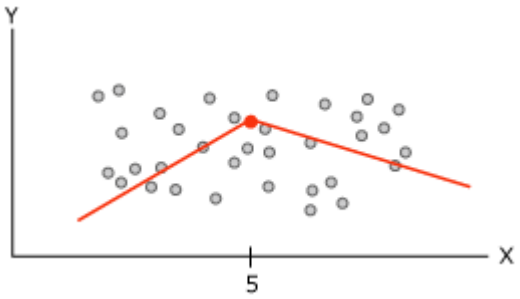
La correlación que aparece en el diagrama hará que el algoritmo de árboles de decisión de Microsoft cree un nuevo nodo en el modelo.



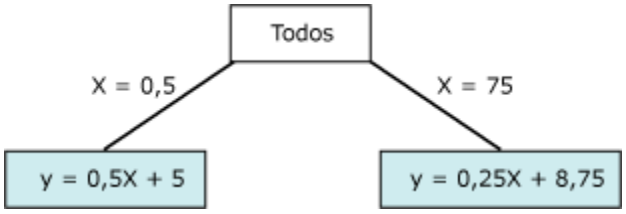
A medida que el algoritmo agrega nuevos nodos a un modelo, se forma una estructura en árbol. El nodo superior del árbol describe el desglose de la columna de predicción para la población global de clientes. A medida que el modelo crece, el algoritmo considera todas las columnas.

Predecir columnas continuas

Cuando el algoritmo de árboles de decisión de Microsoft genera un árbol basándose en una columna de predicción continua, cada nodo contiene una fórmula de regresión. Se produce una división en un punto de no linealidad de la fórmula de regresión. Por ejemplo, considere el siguiente diagrama.



En un modelo de regresión estándar, intentaría derivar una fórmula única que represente la tendencia y las relaciones de los datos como un todo. En cambio, una fórmula única puede hacer un trabajo insuficiente al capturar la discontinuidad en los datos complejos. En su lugar, el algoritmo de árboles de decisión de Microsoft busca los segmentos del árbol que son principalmente lineales y crea fórmulas independientes de estos segmentos. Al dividir los datos en segmentos diferentes, el modelo puede hacer un trabajo mejor de aproximación de datos.
En el siguiente diagrama se representa el diagrama de árbol del modelo en el gráfico de dispersión anterior. Para predecir el resultado, el modelo proporciona dos fórmulas diferentes: una para la bifurcación izquierda, con la fórmula y = .5x x 5 y otra para la bifurcación derecha, con la fórmula y = .25x + 8,75. El punto donde las dos líneas se unen en el gráfico de dispersión es el punto de no linealidad y donde se dividiría un nodo de un modelo de árbol de decisión.



Este es un modelo sencillo con solo dos ecuaciones lineales; por consiguiente, la división en el árbol se encuentra inmediatamente después del nodo **All**. En cambio, una división puede producirse en cualquier nivel del árbol. Eso significa que en un árbol que contenga varios niveles y nodos, donde cada nodo se caracteriza por una colección diferente de atributos, puede que se comparta una fórmula en varios nodos o se aplique solo a un único nodo. Por ejemplo, puede obtener una fórmula de un nodo que se defina como "clientes por encima de una determinada edad e ingresos" y otra en un nodo que represente "clientes que viajan largas distancias". Para ver la fórmula de un nodo o segmento individual, haga clic en el nodo.

Datos requeridos para los modelos de árboles de decisión:

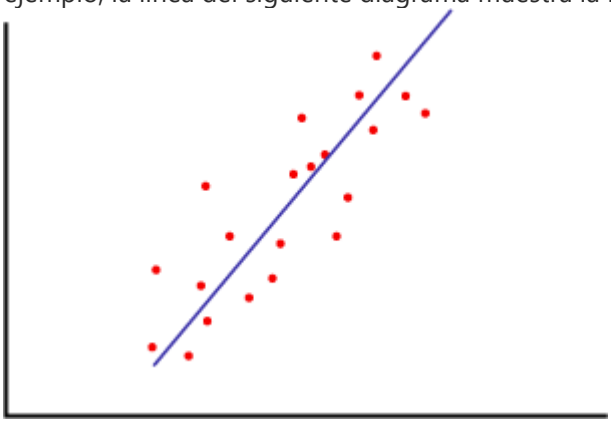
Cuando prepare los datos para su uso en un modelo de árboles de decisión, conviene que comprenda qué requisitos son imprescindibles para el algoritmo concreto, incluidos el volumen de datos necesario y la forma en que estos se utilizan.
Los requisitos para un modelo de árbol de decisión son los siguientes:

- **Una columna de una sola clave** : cada modelo debe contener una columna numérica o de texto que identifique cada registro de manera única. No están permitidas las claves compuestas.

- **Una columna de predicción** . Se requiere al menos una columna de predicción. Puede incluir varios atributos de predicción en un modelo y pueden ser de tipos diferentes, numérico o discreto. Sin embargo, el incremento del número de atributos de predicción puede aumentar el tiempo de procesamiento.
- **Columnas de entrada** . Se requieren columnas de entrada, que pueden ser discretas o continuas. Aumentar el número de atributos de entrada afecta al tiempo de procesamiento.

Algoritmo de regresión lineal:

El algoritmo de regresión lineal de Microsoft es una variación del algoritmo de árboles de decisión de Microsoft que ayuda a calcular una relación lineal entre una variable independiente y otra dependiente y, a continuación, utilizar esa relación para la predicción.
La relación toma la forma de una ecuación para la línea que mejor represente una serie de datos. Por ejemplo, la línea del siguiente diagrama muestra la mejor representación lineal de los datos.



Cada punto de datos del diagrama tiene un error asociado con su distancia con respecto a la línea de regresión. Los coeficientes a y b de la ecuación de regresión ajustan el ángulo y la ubicación de la recta de regresión. Puede obtener la ecuación de regresión ajustando a y b hasta que la suma de los errores asociados a todos los puntos alcance su valor mínimo.
Hay otros tipos de regresión que utilizan varias variables y también hay métodos no lineales de regresión. Sin embargo, la regresión lineal es un método útil y conocido para modelar una respuesta a un cambio de algún factor subyacente.

Ejemplo:

Puede utilizar la regresión lineal para determinar una relación entre dos columnas continuas. Por ejemplo, puede utilizar la regresión lineal para calcular una línea de tendencias en los datos de fabricación o ventas. También podría utilizar la regresión lineal como precursor para el desarrollo de modelos de minería de datos más complejos, con el fin de evaluar las relaciones entre las columnas de datos.

Aunque hay muchas maneras de calcular la regresión lineal que no requieren herramientas de minería de datos, la ventaja de utilizar el algoritmo de regresión lineal de Microsoft para esta tarea es que se calculan y se prueban automáticamente todas las posibles relaciones entre las variables. No tiene que seleccionar un método de cálculo, como por ejemplo para resolver los mínimos cuadrados. Sin embargo, la regresión lineal podría simplificar en exceso las relaciones en escenarios en los que varios factores afectan al resultado.

Cómo funciona el algoritmo?

El algoritmo de regresión lineal de Microsoft es una variación del algoritmo de árboles de decisión de Microsoft . Al seleccionar el algoritmo de regresión lineal de Microsoft , se invoca un caso especial del algoritmo de árboles de decisión de Microsoft , con parámetros que restringen el comportamiento del algoritmo y requieren ciertos tipos de datos de entrada. Además, en un modelo de regresión lineal, el conjunto de datos completo se utiliza para calcular las relaciones en el paso inicial, mientras que en un modelo de árboles de decisión estándar los datos se dividen repetidamente en árboles o subconjuntos más pequeños.

Datos requeridos para los modelos de regresión lineal

Cuando se preparan datos para utilizarse en un modelo de regresión lineal, se deben entender los requisitos del algoritmo determinado. Esto incluye saber cuántos datos se necesitan y cómo se utilizan. Los requisitos para este tipo de modelo son los siguientes:

- **Una columna de una sola clave**: cada modelo debe contener una columna numérica o de texto que identifique cada registro de manera única. No están permitidas las claves compuestas.
- **Una columna de predicción**. Se requiere al menos una columna de predicción. Se pueden incluir varios atributos de predicción en un modelo, pero deben ser tipos de datos numéricos continuos. No se puede utilizar un tipo de datos de fecha y hora como atributo de predicción aunque el almacenamiento nativo para los datos sea numérico.
- **Columnas de entrada** Deben contener datos numéricos continuos y se les debe asignarse el tipo de datos adecuado.

Algoritmo de regresión logística de Microsoft:

La regresión logística es una técnica estadística conocida que se usa para modelar los resultados binarios.
Existen varias implementaciones de regresión logística en la investigación estadística, que utilizan diferentes técnicas de aprendizaje. El algoritmo de Regresión logística de Microsoft se ha implementado utilizando una variación del algoritmo de Red neuronal de Microsoft . Este algoritmo comparte muchas de las cualidades de las redes neurales pero es más fácil de entrenar.

Una de las ventajas de la regresión logística es que el algoritmo es muy flexible, puede tomar cualquier tipo de entrada y admite varias tareas analíticas diferentes:

- Usar datos demográficos para realizar predicciones sobre los resultados, como el riesgo de contraer una determinada enfermedad.
- Explorar y ponderar los factores que contribuyen a un resultado. Por ejemplo, buscar los factores que influyen en los clientes para volver a visitar un establecimiento.
- Clasificar los documentos, el correo electrónico u otros objetos que tengan muchos atributos.

Ejemplo

Imagine un grupo de personas que comparten información demográfica parecida y que adquieren productos de la empresa Adventure Works. Al modelar los datos para relacionarlos con un resultado concreto, como la compra de un producto de destino, podrá ver cómo contribuye la información demográfica a la probabilidad de que alguien adquiera dicho producto de destino.

Cómo funciona el algoritmo

La regresión logística es un método estadístico conocido que se usa para determinar la contribución de varios factores a un par de resultados. La implementación de Microsoft usa una red neuronal modificada para modelar las relaciones entre las entradas y los resultados. Se mide el efecto de cada entrada en el resultado y se ponderan las diversas entradas en el modelo acabado. El nombre regresión logística procede del hecho de que la curva de los datos se comprime mediante una transformación logística para minimizar el efecto de los valores extremos.

Datos requeridos para los modelos de regresión logística:

Al preparar los datos para su uso en el entrenamiento de un modelo de regresión logística, conviene comprender qué requisitos son imprescindibles para el algoritmo concreto, incluidos el volumen de datos necesario y la forma en que estos datos se utilizan.
Los requisitos para un modelo de regresión logística son los siguientes:

Una columna de una sola clave: cada modelo debe contener una columna numérica o de texto que identifique cada registro de manera única. No están permitidas las claves compuestas.

Columnas de entrada: cada modelo debe tener al menos una columna de entrada que contenga los valores que se utilizan como factores en el análisis. Puede tener tantas columnas de entrada como desee, pero dependiendo del número de valores existentes en cada columna, la adición de columnas adicionales podría aumentar el tiempo necesario para entrenar el modelo.

Al menos una columna de predicción: el modelo debe contener al menos una columna de predicción de cualquier tipo de datos, incluidos datos numéricos continuos. Los valores de la columna de predicción también se pueden tratar como entradas del modelo, o se puede especificar que solo se utilicen para las predicciones. No se admiten tablas anidadas en las columnas de predicción, pero se pueden usar como entradas.

Algoritmo Bayes Naive de Microsoft:

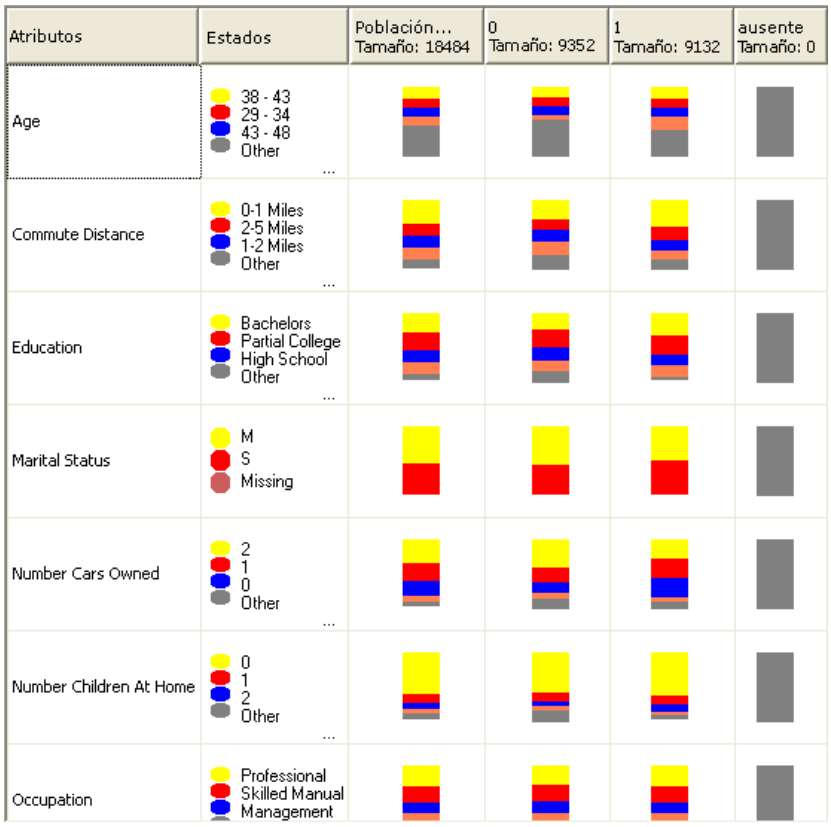
El algoritmo Bayes naive de Microsoft es un algoritmo de clasificación basado en los teoremas de Bayes y se puede usar para el modelado de predicción y de exploración. La palabra naïve (ingenuo en inglés) del término Bayes naive proviene del hecho que el algoritmo utiliza técnicas Bayesianas pero no tiene en cuenta las dependencias que puedan existir.
Desde el punto de vista computacional, el algoritmo es menos complejo que otros algoritmos de Microsoft y, por tanto, resulta útil para generar rápidamente modelos de minería de datos que detectan las relaciones entre las columnas de entrada y las columnas de predicción. Puede utilizar este algoritmo para realizar la exploración inicial de los datos y, más adelante, aplicar los resultados para crear modelos de minería de datos adicionales con otros algoritmos más complejos y precisos desde el punto de vista computacional.

Ejemplo

Como parte de su estrategia promocional, el departamento de comercialización de la empresa Adventure Works Cycles ha decidido atraer a posibles clientes realizando un envío por correo de folletos. Para reducir costos, desean enviar los folletos solo a los clientes de los que esperan recibir respuesta. La empresa almacena información en una base de datos sobre datos demográficos y respuestas a envíos de correo anteriores. Desean utilizar estos datos para ver el modo en que los datos demográficos como la edad o la ciudad pueden ayudarles a predecir la respuesta a una promoción, comparando los clientes potenciales con los que tienen características similares y con los que han adquirido productos de la empresa en el pasado. En concreto, lo que desean es ver las diferencias entre los clientes que adquirieron una bicicleta y los que no lo hicieron.
Mediante el algoritmo Bayes naive de Microsoft , el departamento de comercialización pude predecir rápidamente un resultado de un perfil de cliente concreto y, por tanto, puede determinar qué clientes responderán a los folletos con más probabilidad. Con el Visor Bayes naive de Microsoft de SQL Server Data Tools (SSDT), también pueden investigar visualmente qué columnas de entrada específicas contribuyen a conseguir respuestas positivas a los folletos.

Cómo funciona el algoritmo:

El algoritmo Bayes naive de Microsoft calcula la probabilidad de cada estado de cada columna de entrada, dado cada posible estado de la columna de predicción.
Para comprender cómo funciona, utilice el Visor Bayes naive de Microsoft de SQL Server Data Tools (SSDT) (como se muestra en el siguiente gráfico) para consultar una representación visual del modo en que el algoritmo distribuye los estados.



Aquí, el Visor Bayes naive de Microsoft muestra cada columna de entrada del conjunto de datos e indica cómo se distribuyen los estados de cada columna, dado cada estado de la columna de predicción.
Esta vista del modelo se utilizaría para identificar las columnas de entrada que son importantes para diferenciar los distintos estados de la columna de predicción.
Por ejemplo, en la fila Commute Distance que se muestra aquí, la distribución de valores de entrada es visiblemente diferente para los compradores en comparación con los no compradores. Esto indica que la entrada, Commute Distance = 0-1 miles, es un factor de predicción potencial.
El visor también proporciona valores para las distribuciones, de modo que pueda ver que para los clientes que viajan entre una y dos millas para ir a trabajar, la probabilidad de que compren una bicicleta es de 0,387, y la probabilidad que no la compren es de 0,287. En este ejemplo, el algoritmo utiliza la información numérica, derivada de un dato de cliente (como la distancia entre el domicilio y el lugar de trabajo), para predecir si un cliente compraría una bicicleta.

.

Datos requeridos para los modelos Bayes naive

Al preparar los datos para su uso en un modelo de entrenamiento Bayes naive, conviene comprender qué requisitos son imprescindibles para el algoritmo, incluidos el volumen de datos necesario y la forma en que estos datos se utilizan.
Los requisitos para un modelo Bayes naive son los siguientes:

- **Una columna de una sola clave** : cada modelo debe contener una columna numérica o de texto que identifique cada registro de manera única. No están permitidas las claves compuestas.

- **Columnas de entrada**: en un modelo Bayes naive, todas las columnas deben ser discretas o se deben haber discretizado los valores. Para más información sobre cómo discretizar columnas (bin).

- **Las variables deben ser independientes.** En un modelo Bayes naive, también es importante asegurarse de que los atributos de entrada sean independientes unos de otros. Esto es particularmente importante al utilizar el modelo para la predicción. Si usa dos columnas de datos que ya están estrechamente relacionadas, el efecto sería multiplicar la influencia de esas columnas, lo que puede ocultar otros factores que influyen en el resultado.
  Al contrario, la capacidad del algoritmo de identificar las correlaciones entre las variables es útil cuando está explorando un modelo o conjunto de datos, para identificar las relaciones entre las entradas.

- **Al menos una columna de predicción**: Él atributo de predicción debe contener valores discretos o discretizados. Los valores de la columna predecible se pueden tratar como entradas. Este ejercicio puede ser útil si explora un nuevo conjunto de datos, para encontrar relaciones entre las columnas.

Algoritmo de red neuronal de Microsoft:

El algoritmo de red neuronal de Microsoft es una implementación de la popular arquitectura de red neuronal adaptable para el aprendizaje automático. El algoritmo prueba cada posible estado del atributo de entrada con cada posible estado del atributo de predicción, y calcula las probabilidades de cada combinación según los datos de aprendizaje. Puede usar estas probabilidades para tareas de clasificación o regresión, así como para predecir un resultado en función de algunos atributos de entrada. También se puede usar una red neuronal para el análisis de asociación.
Cuando se crea un modelo de minería de datos con el algoritmo de red neuronal de Microsoft, puede incluir varias salidas y el algoritmo creará varias redes. El número de redes incluidas en un modelo de minería de datos depende del número de estados (o valores de atributo) de las columnas de entrada, así como del número de columnas de predicción que usa el modelo de minería de datos y el número de estados de dichas columnas.

Ejemplo:

El algoritmo de red neuronal de Microsoft es útil para analizar datos de entrada complejos, como los datos de un proceso comercial o de producción, o problemas empresariales para los que hay una cantidad importante de datos de entrenamiento disponibles pero en los que no es fácil derivar reglas mediante otros algoritmos.
Los casos sugeridos para utilizar el algoritmo de red neuronal de Microsoft son:

- Análisis de comercialización y promoción, como medir el éxito de una promoción por correo directo o una campaña publicitaria en la radio.
- Predecir los movimientos de las acciones, la fluctuación de la moneda u otra información financiera con gran número de cambios a partir de los datos históricos.
- Analizar los procesos industriales y de producción.
- Minería de texto.
- Cualquier modelo de predicción que analice relaciones complejas entre muchas entradas y relativamente pocas salidas.

Cómo funciona el algoritmo:

El algoritmo de red neuronal de Microsoft crea una red formada por hasta tres niveles de nodos (en ocasiones denominados *neuronas*). Estos niveles son el *nivel de entrada*, el *nivel oculto* y el *nivel de salida*.

Nivel de entrada: los nodos de entrada definen todos los valores de atributos de entrada para el modelo de minería de datos, así como sus probabilidades.

Nivel oculto: los nodos ocultos reciben entradas de los nodos de entrada y proporcionan salidas a los nodos de salida. El nivel oculto es donde se asignan pesos a las distintas probabilidades de las entradas. Un peso describe la relevancia o importancia de una entrada determinada para el nodo oculto. Cuanto mayor sea el peso asignado a una entrada, más importante será el valor de dicha entrada. Los pesos pueden ser negativos, lo que significa que la entrada puede desactivar, en lugar de activar, un resultado concreto.

Nivel de salida: los nodos de salida representan valores de atributo de predicción para el modelo de minería de datos.
Para obtener una explicación detallada sobre cómo se construyen y puntúan los niveles de entrada, los niveles de salida y los niveles ocultos.

Datos requeridos para los modelos de red neuronal:

El modelo de red neuronal debe contener una columna de clave, una o más columnas de entrada y una o más columnas de predicción.
Los modelos de minería de datos que usan el algoritmo de red neuronal de Microsoft están muy influenciados por los valores que se especifican en los parámetros disponibles para el algoritmo. Los parámetros definen cómo se muestrean los datos, cómo se distribuyen o cómo se espera que estén distribuidos en cada columna, y cuándo se invoca la selección de características para limitar los valores usados en el modelo final.

Algoritmo de clústeres de secuencia de Microsoft:

El algoritmo de clústeres de secuencia de Microsoft es un algoritmo único que combina el análisis secuencial con la agrupación en clústeres. Puede usar este algoritmo para explorar datos que contienen eventos que pueden vincularse con rutas o *secuencias*. El algoritmo encuentra las secuencias más comunes y realiza una agrupación en clústeres para buscar secuencias que sean similares. En los ejemplos siguientes se muestran los tipos de secuencia que se pueden capturar como datos para el aprendizaje automático con el fin de proporcionar información sobre problemas comunes o escenarios empresariales:

- Secuencias de clics o rutas de clics generadas cuando un usuario navega o examina un sitio web.
- Registros que enumeran eventos que preceden a un incidente, como errores de disco duro o interbloqueos del servidor.
- Registros de transacciones que describen el orden en el que un cliente agrega elementos a un carro de la compra en línea.
- Registros que siguen las interacciones del cliente (o paciente) a lo largo del tiempo para predecir cancelaciones del servicio u otros resultados poco satisfactorios.

Este algoritmo es similar en muchas maneras al algoritmo de clústeres de Microsoft . Sin embargo, en lugar de encontrar clústeres de casos que contienen atributos similares, el algoritmo de clústeres de secuencia de Microsoft encuentra clústeres de casos que contienen rutas similares en una secuencia.

Ejemplo

El sitio web de Adventure Works Cycles web site collects information about what pages site users visit, and about the order in which the pages are visited. Debido a que la empresa ofrece un sistema de pedidos en línea, los clientes deben registrarse en el sitio. Esto permite que la empresa pueda conseguir información de clics por cada perfil de cliente. Mediante el uso del algoritmo de clústeres de secuencia de Microsoft en estos datos, la empresa puede encontrar grupos, o clústeres, de los clientes que tienen patrones o secuencias de clics similares. La empresa puede usar estos clústeres para analizar la forma en que los clientes se mueven por el sitio web, identificar qué páginas se relacionan más estrechamente con la venta de un producto en particular y predecir las páginas que tienen mayores probabilidades de ser visitadas a continuación.

Cómo funciona el algoritmo:

El algoritmo de clústeres de secuencia de Microsoft es un algoritmo híbrido que combina técnicas de agrupación en clústeres con el análisis de cadenas de Markov para identificar los clústeres y sus secuencias. Una de las marcas distintivas del algoritmo de clústeres de secuencia de Microsoft es que utiliza los datos de las secuencias. Estos datos suelen representar una serie de eventos o transiciones entre los estados de un conjunto de datos, como una serie de compras de productos o los clics en web para un usuario determinado. El algoritmo examina todas las probabilidades de transición y mide las diferencias, o las distancias, entre todas las posibles secuencias del conjunto de datos con el fin de determinar qué secuencias es mejor utilizar como entradas para la agrupación en clústeres. Cuando el algoritmo cree la lista de secuencias candidatas, usará la información de las secuencias como entrada para el método EM (maximización de la expectativa) de agrupación en clústeres.
Para obtener una descripción detallada de la implementación.

Datos requeridos para los modelos de clústeres de secuencias

Al preparar los datos para usarlos en el entrenamiento de un modelo de agrupación en clústeres de secuencia, conviene comprender qué requisitos son imprescindibles para el algoritmo concreto, incluidos el volumen de datos necesario y la forma en que se usan los datos.
Los requisitos de un modelo de agrupación en clústeres de secuencia son los siguientes:

- **Una columna de clave única** Un modelo de agrupación en clústeres de secuencia necesita una clave que identifique los registros.

- **Una columna de secuencia**: para los datos de la secuencia, el modelo debe tener una tabla anidada que contenga una columna de identificador de secuencia. El id. de secuencia puede ser cualquier tipo de datos ordenable. Por ejemplo, puede usar el identificador de una página web, un número entero o una cadena de texto, con tal de que la columna identifique los eventos en una secuencia. Solo se admite un identificador de secuencia por cada secuencia y un tipo de secuencia en cada modelo.

- **Atributos opcionales no relacionados con la secuencia**: el algoritmo admite la incorporación de otros atributos que no tengan que ver con las secuencias. Estos atributos pueden incluir las columnas anidadas.

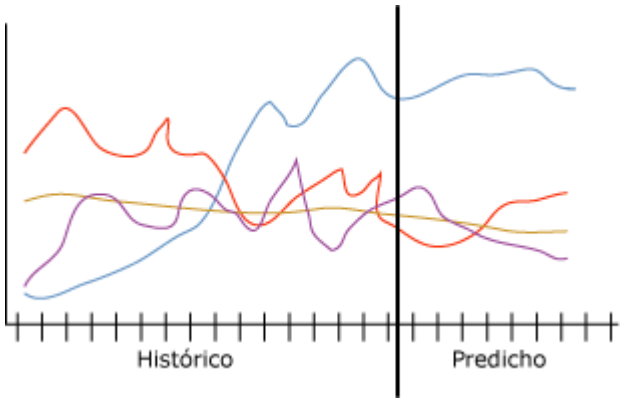
Algoritmo de serie temporal de Microsoft:

El algoritmo de serie temporal de Microsoft proporciona varios algoritmos optimizados para la previsión en el tiempo de valores continuos, tales como las ventas de productos. Mientras que otros algoritmos de Microsoft , como por ejemplo los árboles de decisión, requieren columnas adicionales de nueva información como entrada para predecir una tendencia, los modelos de serie temporal no las necesitan. Un modelo de serie temporal puede predecir tendencias basadas únicamente en el conjunto de datos original utilizado para crear el modelo. Es posible también agregar nuevos datos al modelo al realizar una predicción e incorporar automáticamente los nuevos datos en el análisis de tendencias.
El siguiente diagrama muestra un modelo típico de previsión en el tiempo de las ventas de un producto en cuatro regiones de ventas diferentes. El modelo presentado en el diagrama de ventas muestra las ventas de cada región como líneas de color rojo, amarillo, púrpura y azul. La línea de cada región consta de dos partes:

- La información histórica aparece a la izquierda de la línea vertical y representa los datos que el algoritmo utiliza para crear el modelo.

- La información de la predicción aparece a la derecha de la línea vertical y representa la previsión realizada por el modelo.

A la combinación de los datos de origen y los datos de la predicción se le denomina *serie*.



Una característica importante del algoritmo de serie temporal de Microsoft es su capacidad para llevar a cabo predicciones cruzadas. Si entrena el algoritmo con dos series independientes, pero relacionadas, puede utilizar el modelo generado para predecir el resultado de una serie basándose en el comportamiento de la otra. Por ejemplo, las ventas observadas de un producto pueden influir en las ventas previstas de otro producto. La predicción cruzada también es útil para crear un modelo general que se puede aplicar a múltiples series. Por ejemplo, las predicciones para una región determinada son inestables debido a que la serie no dispone de datos de buena calidad. Podría entrenar un modelo general sobre la media de las cuatro regiones y, a continuación, aplicar el modelo a las series individuales para crear predicciones más estables para cada región.

Ejemplo:

El equipo de administración de Adventure Works Cycles desea predecir las ventas mensuales de bicicletas para el próximo año. La compañía está especialmente interesada en saber si las ventas de un determinado modelo de bicicleta se pueden utilizar para predecir las ventas de otro modelo. Al utilizar el algoritmo de serie temporal de Microsoft en los datos históricos de los últimos tres años, la empresa puede crear un modelo de minería de datos que prevea la venta futura de bicicletas. Además, la organización puede llevar a cabo predicciones cruzadas para ver si las tendencias de venta de modelos individuales de bicicleta están relacionadas.
Cada trimestre, la compañía tiene previsto actualizar el modelo con datos recientes de ventas y actualizar sus predicciones a las tendencias recientes del modelo. Para suplir los datos de los almacenes que no actualizan los datos de ventas de forma precisa o regular, crearán un modelo de predicción general que utilizarán para crear predicciones para todas las regiones.

Cómo funciona el algoritmo:

En Resultado de, el algoritmo de serie temporal de Microsoft usaba un solo método de serie temporal con regresión automática, denominado ARTXP. El algoritmo ARTXP se optimizó para predicciones a corto plazo y, por consiguiente, destacaba en la predicción del siguiente valor probable de una serie. A partir de SQL Server 2008, el algoritmo de serie temporal de Microsoft incluía un segundo algoritmo, ARIMA, optimizado para la predicción a largo plazo. Para obtener una explicación detallada sobre la implementación de los algoritmos ARIMA y ARTXP, De forma predeterminada, el algoritmo de serie temporal de Microsoft utiliza una mezcla de los dos algoritmos al analizar patrones y realizar

predicciones. El algoritmo entrena dos modelos independientes sobre los mismos datos: uno de los modelos usa el algoritmo ARTXP y el otro modelo usa el algoritmo ARIMA. A continuación, el algoritmo combina los resultados de los dos modelos para obtener la mejor predicción sobre un número variable de intervalos de tiempo. Dado que ARTXP obtiene mejores resultados en las predicciones a corto plazo, se le da mayor importancia al principio de una serie de predicciones. Sin embargo, a medida que los intervalos de tiempo que se están prediciendo se adentran en el futuro, se va dando más importancia a ARIMA.
Es posible también controlar la mezcla de algoritmos para favorecer la predicción a corto o a largo plazo en las series temporales. A partir de SQL Server 2008 Standard, puede especificar el algoritmo que se va a usar:

- Utilizar solo ARTXP para la predicción a corto plazo.
- Utilizar solo ARIMA para la predicción a largo plazo.
- Utilizar la mezcla predeterminada de los dos algoritmos.

A partir de SQL Server 2008 Enterprise, también es posible personalizar la manera en que el algoritmo de serie temporal de Microsoft combina los modelos para la predicción. Al utilizar un modelo mixto, el algoritmo de serie temporal de Microsoft combina los dos algoritmos de la manera siguiente:

- Solo ARTXP se utiliza siempre para realizar el primer par de predicciones.
- Tras el primer par de predicciones, se utiliza una combinación de ARIMA y ARTXP.
- A medida que el número de pasos de la predicción aumenta, las predicciones se basan en mayor medida en ARIMA hasta que llega un momento en que ARTXP deja de utilizarse.
- Es posible controlar el punto de combinación, esto es, el ritmo al que la ponderación de ARTXP disminuye y la ponderación de ARIMA aumenta, mediante el parámetro PREDICTION_SMOOTHING.

Ambos algoritmos pueden detectar estacionalidad en los datos en varios niveles. Por ejemplo, sus datos podrían contener ciclos mensuales anidados en ciclos anuales. Para detectar estos ciclos estacionales, es posible proporcionar una sugerencia de periodicidad o bien especificar que el algoritmo deberá detectar automáticamente la periodicidad.
Además de la periodicidad, hay otros parámetros que controlan el comportamiento del algoritmo de serie temporal de Microsoft cuando éste detecta la periodicidad, realiza predicciones o analiza casos.

Datos requeridos para los modelos de serie temporal:

Al preparar los datos para el entrenamiento de cualquier modelo de minería de datos, es preciso comprender los requisitos del modelo en particular así como la forma en que se utilizan los datos.
Cada modelo de previsión debe contener una serie de casos, que es la columna que especifica los intervalos de tiempo u otras series sobre las que se produce el cambio. Por ejemplo, los datos del anterior diagrama muestran las series correspondientes al historial y a la previsión de ventas de bicicletas para un período de varios meses. Para este modelo, cada región es una serie y la columna de fecha contiene la serie temporal, que también es la serie de casos. En otros modelos, la serie de escenarios puede ser un campo de texto o algún identificador tal como un id. de cliente o de transacción. Sin embargo, un modelo de serie temporal debe siempre utilizar una fecha, una hora o algún otro valor numérico único para su serie de escenarios.
Los requisitos para un modelo de serie temporal son los siguientes:

- **Una única columna Key Time** Cada modelo debe contener una columna numérica o de fecha que se utilizará como serie de casos y que define los intervalos de tiempo que utilizará el modelo. El tipo de datos para la columna de clave temporal puede ser un tipo de datos datetime o bien numérico. Sin embargo, la columna debe contener valores continuos y éstos deben ser únicos para cada serie. La serie de casos para un modelo de serie temporal no pueden estar almacenada en dos columnas como por ejemplo una columna Año y una columna Mes.

- **Una columna predecible** Cada modelo debe contener por lo menos una columna predecible alrededor de la que el algoritmo generará el modelo de serie temporal. El tipo de datos de la columna predecible debe contener valores continuos. Por ejemplo, es posible predecir la manera en que los atributos numéricos tales como ingreso, ventas o temperatura, varían con el tiempo. Sin embargo, no es posible utilizar como columna predecible una columna que contenga valores discretos tales como el estado de las compras o el nivel de educación.

- **Una columna de clave de serie opcional** Cada modelo puede tener una columna de clave adicional que contenga valores únicos que identifiquen a una serie. La columna de clave de serie opcional debe contener valores únicos. Por ejemplo, un solo modelo puede contener ventas de muchos modelos de producto, siempre y cuando haya un solo registro para cada nombre del producto para cada intervalo de tiempo.

Puede definir los datos de entrada para el modelo de serie temporal de Microsoft de dos formas. Sin embargo, puesto el formato de los escenarios de entrada afecta a la definición del modelo de minería, debe considerar sus necesidades de negocio y preparar sus datos en consecuencia. Los dos ejemplos siguientes muestran cómo los datos de entrada afectan al modelo. En ambos ejemplos, el modelo de minería completado contiene patrones de cuatro series distintas:

.

- Ventas para el producto A
- Ventas para el producto B
- Volumen para el producto A
- Volumen para el producto B

En ambos ejemplos, puede predecir nuevas ventas futuras y volúmenes para cada producto. No puede predecir nuevos valores para el producto o para el tiempo.

TimeID	Product	Sales	Volume
1/2001	A	1000	600
2/2001	A	1100	500

Ejemplo 1: Conjunto de datos de serie temporal con serie representada como valores de columna

En este ejemplo se utiliza la siguiente tabla de escenarios de entrada:

1/2001	B	500	900
2/2001	B	300	890

La columna TimeID de la tabla contiene un identificador de tiempo e incluye dos entradas para cada día. La columna TimeID se convierte en la serie de casos. Por consiguiente, esta columna se designaría como la columna de clave temporal para el modelo de serie temporal.
La columna Product define un producto de la base de datos. Esta columna contiene la serie del producto. Por consiguiente, esta columna se designaría como una segunda clave para el modelo de serie temporal.
La columna Sales describe los beneficios brutos del producto especificado para un día y la columna Volume describe la cantidad del producto especificado que permanece en el almacén. Estas dos columnas contienen los datos que se utilizan para entrenar el modelo. Los atributos Sales y Volume pueden ser atributos de predicción para cada serie de la columna Product.

Ejemplo 2: Conjunto de datos de serie temporal con cada serie en una columna independiente

Aunque en este ejemplo se utilizan básicamente los mismos datos de entrada que en el primer ejemplo, estos se estructuran de manera diferente, como se muestra en la siguiente tabla:

TimeID	A_Sales	A_Volume	B_Sales	B_Volume
1/2001	1000	600	500	900
2/2001	1100	500	300	890

En esta tabla, la columna TimeID contiene todavía la serie de casos para el modelo de la serie temporal que fue designada como la columna de clave temporal. Sin embargo, las antiguas columnas de ventas y volumen están ahora divididas en dos columnas, cada una de las cuales va precedida por el nombre del producto. Como resultado, solo existe una única entrada para cada día en la columna TimeID. Se

crea así un modelo de serie temporal que contendría cuatro columnas predecibles: A_Sales, A_Volume, B_Sales y B_Volume.
Además, puesto que los productos se han distribuido en columnas diferentes, no es preciso especificar una columna de clave de serie adicional. Todas las columnas del modelo son o una columna de serie de casos o bien una columna predecible.

Conclusión

Los algoritmos de Minería de datos son bastantes útiles para la toma de decisiones en una empresa, todos los algoritmos de minería de datos se pueden personalizar ampliamente y usar mediante programación con las API proporcionadas. También puede automatizar la creación, aprendizaje y reciclaje de modelos con los componentes de minería de datos de Integration Services. Además, puede usar algoritmos de minería de datos desarrollados por terceros que cumplan con la especificación OLE DB para minería de datos, o bien desarrollar algoritmos personalizados que se puedan registrar como servicios para usarlos después en el marco de la minería de datos de SQL Server.

Los algoritmos proporcionados en la minería de datos de SQL Server son los métodos más comunes y probados para derivar patrones a partir de datos. Por ejemplo, la agrupación en clústeres mediana-K es uno de los algoritmos de agrupación en clústeres más antiguo y está disponible en un gran número de herramientas y con diferentes implementaciones y opciones.