INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

TECNOLÓGICO DE MONTERREY®

GENERACIÓN DE MODELOS PREDICTIVOS DE SATISFACCIÓN TRANSACCIONAL PARA UN CENTRO DE ATENCIÓN A CLIENTES.

TESIS QUE PARA OPTAR EL GRADO DE MAESTRO EN CIENCIAS COMPUTACIONALES CON ESPECIALIDAD EN REDES Y SEGURIDAD INFORMÁTICA PRESENTA

CÉSAR SALAZAR HERNÁNDEZ

Asesor: Dr. MIGUEL GONZÁLEZ MENDONZA

Jurado:	Dra. MARÍA DE LOS ÁNGELES JUNCO REY	Presidente
	Dr. JORGE ADOLFO RÁMIREZ URESTI	Secretario
	Dr. MIGUEL GONZÁLEZ MENDOZA	Vocal

Atizapán de Zaragoza, Edo. Méx., Mayo de 2012.

DEDICATORIA

A mis padres, por todo su el apoyo, consejos, motivación. Sin ustedes, este trabajo no hubiera sido posible.

RECONOCIMIENTOS

Agradezco sinceramente al Consejo Nacional de Ciencia y Tecnología (CONACYT), por el apoyo brindado para la realización de mis estudios del grado de maestría.

De igual forma, al Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), campus Estado de México, por las facilidades brindadas para la realización de mis estudios en tan prestigiosa institución educativa.

Así mismo, agradezco a la Dra. María de los Ángeles Junco Rey y al Dr. Jorge Ramírez Uresti por su participación como miembros del comité de defensa de tesis.

Finalmente, mi más profundo agradecimiento a mi asesor de tesis, Dr. Miguel González Mendoza, por su tiempo, motivación y orientación durante el desarrollo de este trabajo.

RESUMEN

Es una necesidad de las empresas lograr que sus clientes estén altamente satisfechos con los servicios que estas les proveén. Un cliente insatisfecho cambiará de proveedor con mayor facilidad que un cliente que tiene un alto grado de satifacción con su proveedor actual y la perdida de clientes repercute directamente en las finanzas de la empresa, por lo que es importante crear mecanismos que ayuden a mejorar las relaciones cliente – proveedor. Existen herramientas y estrategias que constribuyen para este fin, como los CRM, programas de recompensas, etc.

No hay que olvidar que la satisfacción se construye mediante experiencias. Cada contacto entre cliente y proveedor, ya sea para solicitar un servicio o levantar una queja son puntos de crucial importancia para construir un índice de satisfacción cliente, por lo que es muy importante detectar cuando uno de estos contactos o transacciones está siendo una experiencia insatisfactoria para el cliente o tiene una tendencia clara a serlo, para tomar las medidas pertinentes que permitan modificar dicha tendencia.

Es así que esta tesis surge de la necesidad de predecir con un alto grado de exactitud qué transacciones en un centro de atención teléfonica a clientes tienen una tendencia a ser consideradas una mala experiencia para el cliente, contribuyendo a que se sienta insatisfecho, y describir qué características de la transacción determinan esa tendencia.

Este trabajo se centra el uso de minería de datos para la construcción de clasificadores, usando la técnica de árboles de decisión con el algoritmo C4.5. Este algoritmo fue seleccionado debio a su amplio uso en la literatura y a que el resultado es fácilmente interpretable en comparación con modelos como Redes neuronales o Máquinas de soporte de vectores.

Los resultados obtenidos fueron positivos, ya que se logró una precisión de más del 78% en la predicción de casos con tendencia a ser satisfactorios contra casos con tendencia a ser instisfactorios. Adicionalmente el clasificador generado tiene un tamaño adecuado para ser interpretado por el personal de atención y directivos del centro de llamadas, ayudándoles a tomar decisiones que conduzcan a mejoras sustanciales en el servicio que proveen.

Durante el desarrollo de este trabajo se logró un profundo entendimiento del negocio lo que permite sugerir áreas de investigación que complementen este trabajo, mejorando los resultados obtenidos y logrando aún mejores índices de satisfacción entre los clientes de esta empresa.

CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

1 INTRODUCCIÓN

Las empresas actuales, incluyendo a aquellas dedicadas a la provisión de servicios de tecnología, se encuentran inmersas en un ambiente altamente competitivo, en el que tienen que enfrentarse a competidores no sólo locales, sino globales. Este nivel de competencia favorece la migración de clientes entre proveedores de bienes o servicios similares, y dado que los beneficios de una empresa están directamente relacionados con el número de clientes que esta posee [1], es una necesidad de las industrias aumentar la cantidad de clientes con los que se cuentan mediante estrategias de captación, al mismo tiempo que se debe disminuir la tasa de migraciones, mediante estrategias de retención.

Existen diversas razones por las que un cliente puede cambiar de proveedor de servicios, entre ellas motivos económicos, geográficos, estratégicos, etc. Sin embargo, uno de los principales motivos de cambio es la insatisfacción que un cliente experimenta con los bienes o servicios que su proveedor le otorga.

Si bien la satisfacción del cliente es una medida perceptual, existen algunas métodos para asignarles un valor objetivo, manipulable y útil para la toma de decisiones. Las encuestas de satisfacción son uno de los métodos de medición de la satisfacción más usados. Estos ejercicios pueden realizarse después de una transacción cliente – proveedor, por ejemplo, después de una compra vía telefónica, el comprador puede indicar que tan satisfecho se encuentra con la atención recibida por parte del vendedor; o pueden realizarse después de un periodo de tiempo en el que pueden llevarse a cabo un gran número de transacciones, por ejemplo, mediante encuestas anuales de satisfacción. La satisfacción global de un cliente puede caracterizarse también como una construcción acumulativa de todas las transacciones o experiencias entre cliente y proveedor.

Un problema de las encuestas de satisfacción es que estas son posteriores a la provisión del bien o servicio, de tal manera que es difícil tomar medidas durante la transacción para prevenir que esta resulte en una experiencia no satisfactoria para el cliente.

El presente trabajo tiene como objetivo desarrollar un modelo que sea capaz de predecir con un alto grado de exactitud qué transacciones en un centro de atención telefónica resultarán insatisfactorias, al mismo tiempo indiquen cuáles son los atributos de la transacción que determinan el resultado de satisfacción o insatisfacción de la transacción, de tal manera que sea posible tomar medidas preventivas durante el desarrollo de la misma con el fin disminuir el grado de insatisfacción de los clientes.

Este documento está estructurado de la siguiente manera: En el resto de este capítulo se brindan antecedentes sobre los conceptos que se manejarán durante el desarrollo de esta tesis y se proporciona la justificación y los objetivos que se pretenden alcanzar con este trabajo. En el segundo capítulo se da una introducción a la teoría de minería de datos, algoritmos de minería y de datos y específicamente el algorito C4.5, y se da un recorrido por los trabajos relacionados y estado del arte. En el tercer capítulo se centra en la creación de los modelos predictivos usando la información de las transacciones del cliente, haciendo énfasis en las etapas de obtención de datos, limpieza de datos y generación de clasificadores. En el cuarto capítulo se mostrarán y analizarán los resultados obtenidos. Finalmente, en el quinto capítulo se señalan las conclusiones del trabajo y se establecen directrices de trabajo futuro.

1.1 ANTECEDENTES

1.1.1 SATISFACCIÓN DEL CLIENTE

En términos generales, la satisfacción se describe como la actitud que una persona tiene hacia un aspecto de su entorno. En el caso de un cliente que recibe un servicio por parte de una empresa, su satisfacción es la actitud que presenta hacia dicho servicio [2].

El grado de satisfacción de un cliente es una indicador que varía con el tiempo, y que depende, según el paradigma de "Confirmación / Desconfirmación" propuesto por Davis y Heineke [3], de la diferencia entre las expectativas del cliente hacia un producto o servicio y el desempeño del mismo. Si las expectativas del cliente son cubiertas por el desempeño real, el cliente estará satisfecho; en otro caso, el cliente estará insatisfecho con el producto o servicio recibido. El modelo de satisfacción de Locke agrega una variable importancia para calificar la satisfacción de un cliente, qué según Kanning [2] no provee mejores indicios del grado de satisfacción que si únicamente se consideran las variales expectativas y desempeño.

Algunos trabajos han demostrado que el grado de satisfacción de un cliente es un indicador que está ligado directamente con el grado de rentabilidad de una empresa. De acuerdo a un estudio de Anderson [4], una mejora de 1% en el grado de satisfacción de los clientes de una empresa conduce a un incremento en el Retorno de Inversión de 2.37%, mientras que Rucci [5] estudió una serie de cambios que se dieron en la cadena de tiendas Sears, desarrollando un modelo que indica que un a partir de un plan de mejora en la actitud de los dependientes, la satisfacción de los clientes incrementa en 1.3% y este aumento conduce a una mejoría de 0.5% en los ingresos de la compañía.

Como se mencionó anteriormente, la insatisfacción de los clientes es uno de los principales motivos por los que estos cambian de proveedor de bienes o servicios, afectando las ganancias de las empresas que pierden clientes. Adicionalmente, algunos estudios señalan que es más costoso para una empresa ganar nuevo clientes que conservar los que ya tiene.

Por estos motivos, para una empresa es importante contar con mecanismos que permitan detectar cuáles de sus clientes están insatisfechos o tienen tendencias a estar insatisfechos en un periodo de tiempo cercano, y con esta información, prevenir el abandono y por ende, la disminución de ganancias.

1.1.2 MESA DE SERVICIO

Según el marco de referencia ITIL [6], el Service Desk o Mesa de servicio actúa como punto único de contacto para cubrir las necesidades de los usuarios de los servicios de TI y de los especialistas que proveen estos servicios. De esta manera, se pretende restaurar, en caso de que se presente una falla o degradación, los servicios a un nivel normal de operaciones con un impacto mínimo a la operación del cliente.

La mesa de servicio tiene dos funciones principales:

- Recibe reportes de incidentes en los servicios (Fallas, interrupciones, retrasos)

- Recibe solicitudes de servicios (*Service request*) de los clientes (Servicios esporádicos, bajo demanda)

Existe diversos tipos de Mesa de ayuda:

- Centro de llamadas, que únicamente tiene la capacidad para recibir y redirigir las llamadas de los clientes a los especialistas técnicos.
- Mesa de ayuda no capacitada, que además de recibir y redirigir estas llamadas, hace un seguimiento de las soluciones y brinda retroalimentación a los clientes.
- Mesa de ayuda capacitada, que recibe y redirige llamadas, hace un seguimiento de las soluciones, brinda retroalimentación a los clientes y puede resolver ciertos incidentes básicos.
- Mesa de ayuda experta, que además de incorporar las capacidades de los tres tipos anteriores, realiza Gestión de Incidentes y Gestión de Problemas.

Una mesa de servicio estándar realiza las siguientes actividades:

- Recibe llamadas, correos electrónicos u otros mensajes que informen sobre incidentes en los servicios que recibe el cliente.
- Llevar un registro de los incidentes reportados.
- Clasificar los incidentes reportados.
- Priorizar los incidentes reportados.
- Escalar los incidentes reportados, es decir, redirigirlos a un especialista para que les brinde solución en el caso de que la persona que recibe el reporte no sea capaz de solucionarlo.
- Registro de soluciones previas a problemas similares.

La siguiente figura ilustra el flujo de actividades estándar que sigue el personal de la mesa de servicio para dar solución a un problema.

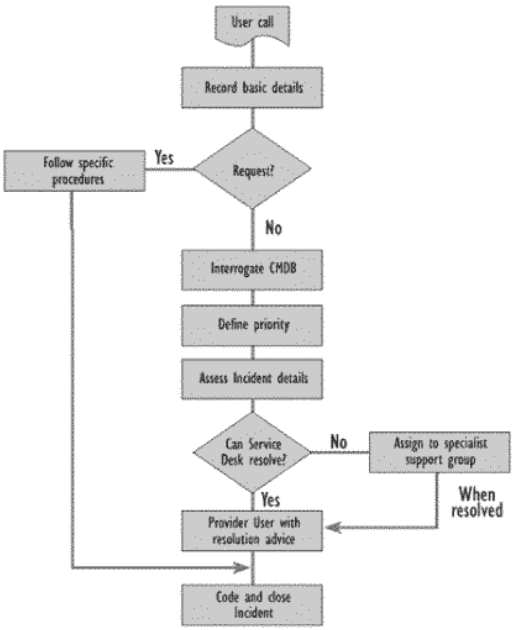


Figura 1.1. Flujo de actividades en la Mesa de Servicio

1.1.3 DESCRIPCIÓN DEL CASO

Para el desarrollo de la presente tesis, se usaron datos provistos Mexis. Mexis es una empresa que brinda servicios administrados en temas de seguridad informática, redes y aplicaciones.
La empresa cuenta con una mesa de servicio que registra vía telefónica los incidentes y solicitudes de servicio que reportan los clientes.

La mesa de servicio tiene la capacitación para resolver problemas básicos usando herramientas informáticas de monitoreo. Si el personal de la mesa de servicio no puede resolver el problema, ya sea porque considera que no tiene la capacitación suficiente o porque el tiempo de atención del incidente o solicitud de servicio ha sobrepasado un límite de tiempo, estos casos se escalan a un nivel de atención superior.

En el segundo nivel de atención, el personal se agrupa en 3 áreas especializadas:

- Centro de operaciones de seguridad (*Security Operations Center* o SOC) que resuelve incidentes y solicitudes de servicio asociadas con la seguridad de la información, monitoreando y configurando equipos tales como Firewalls, Sistemas de detección de intrusos(IDS), Sistemas de prevención de intrusiones (IPS), Proxies y herramientas como antivirus, listas de control de acceso, etc [7].
- Centro de operaciones de redes (*Networking Operations Center* o NOC) que resuelve incidente y solicitudes de servicio asociadas con la infraestructura de redes, monitoreando y configurando equipos tales como Routers, Switches, Enlaces de Red, Servidores DNS.
- Centro de datos (CDC) que resuelve incidente y solicitudes de servicios asociados con al infraestructura de almacenamiento compartido. Esta infraestructura incluye equipos y servicios como: Hospedaje de páginas web, servicio de correo electrónico, servidores web y servidores FTP.

Con el fin de asegurar la calidad de los servicios que provee, la empresa recopila información de diversas fuentes, entre las que se encuentran:

- Registro de llamadas en el centro de llamadas, con indicadores como el tiempo de llamada, el tiempo de espera y número de llamadas por hora.
- Protect System®, una combinación de hardware y software que monitorea de manera remota los dispositivos que el cliente indica para reconocer de manera inmediata fallas en la provisión continua del servicio.
- Sistema de registro de casos de atención, que se explica en la siguiente sección.
- Memorias técnicas de implementación de servicios.
- Base de datos de gestión de configuraciones (*Confuration Managemente Database* o CMDB)
- Base de datos de servicios contratados por cliente.
- Bas de datos de acuerdos de nivel de servicio (*Service Level Agreements* o SLA)
- Base de datos de facturación de clientes por periodo de tiempo.

En la siguiente figura se puede apreciar la integración de los repositorios de información en el proceso de reporte y resolución de problemas dentro de la empresa:

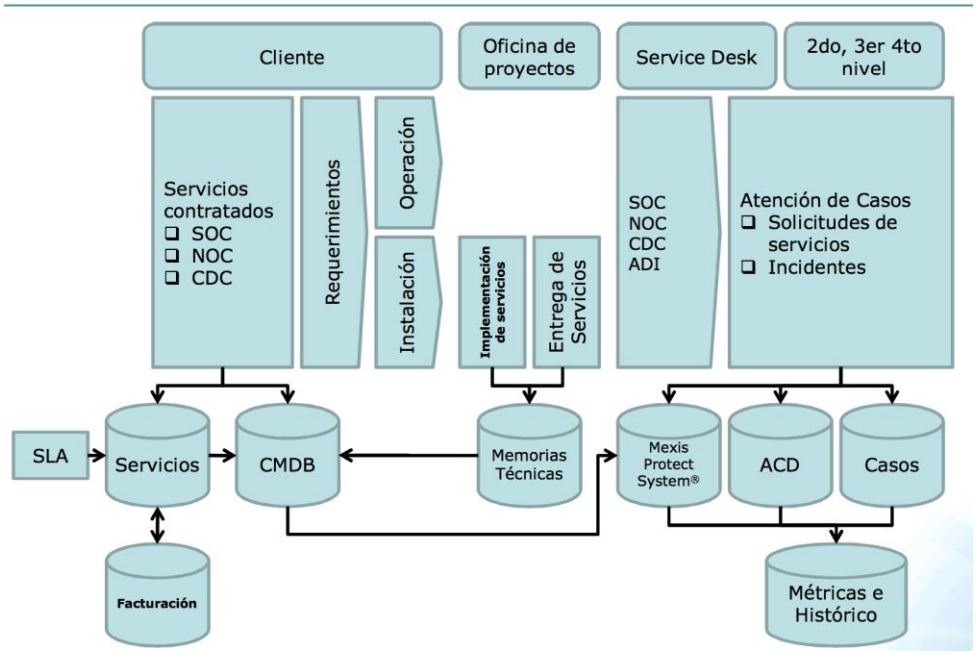


Figura 1.2. Fuentes de información del proceso de resolución de incidentes y solicitudes de servicio

1.1.3.1 Descripción del proceso de registro de casos de atención.

Los reportes de incidentes y solicitudes de servicio son agrupados dentro del término común casos de atención o tickets.

Un caso de atención puede ser iniciado por una llamada del cliente reportando la falla que ha detectado o la solicitud de servicio que necesita, o por el personal de Mexis, que mediante los dispositivos de monitoreo detecta fallas en los equipos del cliente o agrega solicitudes de servicio que considera pueden ser útiles al cliente, por ejemplo, actualizaciones de seguridad en software. Si el caso es iniciado por el cliente, se considera reactivo, mientras que si es iniciado por personal de Mexis, es considerado proactivo.

Tantos los casos reactivos como los proactivos reciben el mismo grado de atención, sin embargo, el proceso que sigue cada clase es ligeramente diferente.

El proceso que sigue un caso de atención reactivo es el siguiente:

1. El cliente realiza una llamada el centro de atención de Mexis (mesa de servicio).
2. El empleado que lo atiende recibe su solicitud de servicio o reporte de falla.
3. El empleado categoriza el caso y lo guarda en el sistema de seguimiento de casos, indicando si el caso corresponde a una falla o solicitud de servicio así como el área especializada que puede atender el caso (SOC, NOC, CDC)
4. Si el empleado es capaz satisfacer la solicitud o resolver la falla, conserva el caso. Si el empleado no cuenta con la capacidad técnica para resolver la falla o solicitud de servicio, envía el caso al área especializada. Si el área especializada no es capaz de resolver el caso, por ejemplo, cuando se trata de fallas que debe solucionar un fabricante de hardware o software, escala dicho caso al fabricante.

5. Una vez que el caso haya sido resuelto, se notifica al cliente para él evalúe la solución e indique si está satisfecho con esta. Si el cliente está satisfecho con la solución, se procede al cierre del caso. Si el cliente no está satisfecho con la solución, se regresa al paso 4.
6. Una vez que se cierra el caso, se envía una encuesta al cliente en el que este indica si estuvo satisfecho con el servicio recibido o no.

Los casos de atención proactivos se diferencia en la entidad que origina el registro del caso, siendo en este caso el personal de Mexis quien dispara el registro. Existe una ligera diferencia en el proceso que sigue los caso de reporte de incidente proactivos y los casos de solicitud de servicio proactivo, principalmente por la motivación que los origina.

El proceso que sigue un caso proactivo para una falla es el siguiente:

1. El personal de Mexis que monitorea los servicios y equipos del cliente detecta una falla.
2. El personal de Mexis registra el incidente en el sistema de seguimiento de casos y lo categoriza, indicando qué tipo de falla ocurre y qué área especializada puede atenderlo.
3. Se notifica al cliente que se detecto una falla en sus servicios o equipos y que se está trabajando para resolverla. La notificación al cliente puede ser previa al registro de la categorización del caso en el sistema de seguimiento.
4. El personal de Mexis transfiere la falla al área especializada para su atención. Se sigue el mismo principio de escalación del caso que en los casos de atención reactivos.
5. Cuando la falla ha sido resuelta, se notifica al cliente para que este evalúe la solución e indique si está satisfecho con esta. Si el cliente esta satisfecho con la solución, el caso se marca como cerrado. Si el cliente no está satisfecho con la solución, se regresa al paso 4.
6. Cuando el caso ha sido marcado como cerrado, se envía una encuesta al cliente para este indique si estuvo satisfecho con el servicio recibido o no.

El proceso que sigue un caso proactivo para una solicitud de servicio es el siguiente:

1. El personal de Mexis detecta un servicio que puede ser de potencialmente útil para el cliente, por ejemplo, una actualización en algún elemento de software.
2. Se notifica al cliente que existe dicho servicio.
3. Si el cliente accede a la ejecución del servicio ofrecido, el personal de Mexis se encarga de llevarlo a cabo, realizando para esto las escalaciones que considere necesarias.
4. Cuando se ha ejecutado el servicio, se notifica al cliente para que este evalúe los resultados de la aplicación del servicio. Si el cliente está satisfecho con los resultados, el caso se marca como cerrado. Si el cliente no está satisfecho con los resultados, se regresa al paso 3.
5. Cuando el caso ha sido marcado como cerrado, se envía al cliente una encuesta para que indique si está satisfecho o no con el servicio proporcionado.

1.1.3.2 Sistema de registro de casos de atención

La empresa lleva el registro de los reportes de incidentes y solicitudes de servicio recibidos, en adelante agrupados bajo el término común casos de atención, mediante un sistema llamado Footprints [8]. FootPrints® almacena los atributos de un caso de atención una base de datos Microsoft SQL Server®.

La información relativa al registro y seguimiento de casos en Footprints está registrada en una tabla denominada vmxTicketDimensions. La tabla proporciona la lista de los campos que componen dicha tabla. Con el fin de proporcionar ejemplos de los valores contenidos en los campos, se usará como referencia el caso con identificador 500000 ingresado el día 31 de octubre de 2011 a las cero horas con 12 minutos y 37 segundos, y cuya última actualización se realizó el día 17 de enero de 2012 a las 11 horas con 9 minutos y 47 segundos.

Tabla 1.1 Campos disponibles en Footprints

Nombre del campo	Descripción del campo	Tipo	Ejemplo
TicketId	El identificador único del caso	Numérico	500000
ProjectId	El identificador del proyecto asociado con el caso.	Numérico	0
EnteredQuarter	El cuarto del año en que se ingresó el caso	Cadena de texto	2011Q4
EnteredPeriod	El periodo en que se ingresó el caso. Se denomina periodo, a la combinación año-mes.	Cadena de texto	2011-10
EnteredDate	La fecha en que se ingresó el caso, colocando la hora, minuto y segundo como 00:00:00	Cadena de texto con formato AAAA-MM-DD hh:mm:ss	2011-10-31 00:00:00
EnteredDateHour	La fecha en que se ingresó el caso, con el valor de la hora adecuado y el valor de minuto y segundo como 00:00	Cadena de texto con formato AAAA-MM-DD hh:mm:ss	2011-10-31 00:00:00
EnteredDateTime	La fecha en que se ingresó el caso, con los valores de hora, minuto y segundo adecuados	Cadena de texto con formato AAAA-MM-DD hh:mm:ss	2011-10-31 00:12:37
EnteredWeekNameId	La concatenación del año y la semana del año en que se ingresó el caso.	Cadena de texto	201145
EnteredWeekName	El día de inicio y fin de la semana en que se ingresó el caso.	Cadena de texto	Oct-31 a Nov-06
LastStatusQuarter	El cuarto del año en	Cadena de texto	2012Q1

	que se realizó la última actualización del caso.		
LastStatusPeriod	El periodo en que se realizó la última actualización del caso. Se denomina periodo, a la combinación año-mes.	Cadena de texto	2012-01
LastStatusDate	La fecha en que se realizó la última actualización del caso, colocando la hora, minuto y segundo como 00:00:00	Cadena de texto con formato AAAA-MM-DD hh:mm:ss	2012-01-17 00:00:00
LastStatusDateHour	La fecha en que se realizó la última actualización del caso, con el valor de la hora adecuado y el valor de minuto y segundo como 00:00	Cadena de texto con formato AAAA-MM-DD hh:mm:ss	2012-01-17 11:00:00
LastStatusDateTime	La fecha en que se realizó la última actualización del caso, con los valores de hora, minuto y segundo adecuados	Cadena de texto con formato AAAA-MM-DD hh:mm:ss	2012-01-17 11:9:47
LastStatusWeekNameId	La concatenación del año y la semana del año en que se realizó la última actualización del caso.	Cadena de texto	201204
LastStatusWeekName	El día de inicio y fin de la semana en que se realizó la última actualización del caso.	Cadena de texto	Ene-16 a Ene-22
CustomerId	El identificador único del cliente que ingresó el caso	Numérico	9128
CustomerName	El nombre del cliente que ingresó el caso	Cadena de texto	Cualquier valor
CustomerShorName	El nombre corto del cliente que ingresó el caso	Cadena de texto	Cualquier valor
GuarantorId	El identificador único del garante asociado con el cliente que	Numérico	9128

	ingresó el caso		
GuarantorName	El nombre del garante asociado con el cliente que ingresó el caso	Cadena de texto	Cualquier valor
GuarantorShortName	El nombre corto del garante asociado con el cliente que ingresó el caso.	Cadena de texto	Cualquier valor
CustomerActive	Indica si un cliente está activo (recibe servicio por parte de Mexis) al momento de la consulta	Carácter	Y (Sí) / N (No)
CustomerStaffId	El identificador de la persona que funge como contacto técnico por parte del cliente	Numérico	5600
CustomerStaffName	El nombre de la persona que funge como contacto técnico por parte del cliente	Cadena de texto	Cualquier valor
CustomerSalesId	El identificador de la persona que funge como contacto de ventas por del cliente	Numérico	5700
CustomerSalesName	El nombre de la persona que funge como contacto de ventas por parte del cliente	Cadena de texto	Cualquier valor
CustomerSegmentoId	El identificador del segmento de mercado al que pertenece el cliente	Numérico	1 (Segmento diamante), 2 (Segmento platino), 3 (Segmento oro), 4 (Segmento plata), 5 (Segmento bronce)
CustomerSegmentoName	El nombre del segmento de mercado al que pertenece el cliente	Cadena de texto	Diamante, Platino, Oro, Plata, Bronce
CustomerSegmentoAtencionId	Identificador del grupo de atención al que pertenece un cliente	Numérico	1 (ODyP) , 2 (PyB)
CustomerSegmentoAtencionName1	Nombre del grupo de atención al que pertenece un cliente	Cadena de texto	ODyP (Oro, Diamante y Platino), PyB (Plata y Bronce)
CustomerRegionId	El identificador de la	Numérico	1

	región geográfica a la que pertenece un cliente		
CustomerRegionName	El nombre de la región geográfica a la que pertenece el cliente	Cadena de texto	México, Puebla
SegmentoMercadoId	El identificador del segmento de mercado al que pertenece el cliente	Numérico	1
SegmentoMercadoName	El nombre del segmento de mercado al que pertenece el cliente	Cadena de texto	Diamante, Platino, Oro, Plata, Bronce
CanalMarketingId	El identificador del canal de ventas mediante el cual el cliente contrató los servicios de Mexis	Numérico	1, 2
CanalMarketingName	El nombre del canal de ventas mediante el cual el cliente contrató los servicios de Mexis	Cadena de texto	AES (Aliado estratégico), Venta directa
AssignedStaffId	El identificador del empleado de Mexis que fue asignado para dar seguimiento al caso	Numérico	1
AssignedStaffName	El nombre del empleado de Mexis que fue asignado para dar seguimiento al caso	Cadena de texto	Cualquier valor
AssignedDireccionId	El identificador de la dirección a la que fue asignado un caso	Numérico	Todos los valores para este campo son 0
AssignedDireccionName	El nombre de la dirección a la que fue asignado un caso	Cadena de texto	Este campo no cuenta con valores
AssignedGerenciaId	El identificador de la gerencia a la que fue asignado un caso	Numérico	Todos los valores para este campo son 0
AssignedGerenciaName	El nombre de la gerencia a la que fue asignado un caso	Cadena de texto	Este campo no cuenta con valores
StatusId	El identificador estado de atención actual del	Numérico	5, 4, etc

	caso		
StatudName	El nombre del estado de atención actual del caso	Cadena de texto	Abierto, Cerrado, En espera
StatusShortName	El nombre corto del estado de atención actual del caso	Cadena de texto	Cls, Opn
AreaId	El identificador del área que registró el caso	Numérico	Todos los valores son 0 para este campo
AreaName1	El nombre corto del área que registró el caso	Cadena de texto	SERVICE DESK, ADI (Administración de incidentes)
AreaName2	El nombre del área que registró el caso	Cadena de texto	SERVICE DESK, ADI
CategoriaId	El identificador de la categoría del caso, de acuerdo al tipo de equipo o servicio que afecta	Numérico	Todos los valores son 0 para este campo
CategoriaName	El nombre de la categoría del caso, de acuerdo al tipo de equipo o servicio que afecta	Cadena de texto	Firewall, Servidor dedicado, etc
AccionId	El identificador del modo de atención del caso	Numérico	1, 2
AccionName	El nombre del modo de atención del caso	Cadena de texto	Proactivo, Reactivo
GrupoAreaId	El identificador del grupo al que pertenece el área que atiende el caso	Numérico	2, 3
GrupoAreaName	El nombre corto del grupo al que pertenece el área que atiende el caso	Cadena de texto	OPE, ATC
GrupoAreaName2	El nombre del grupo al que pertenece el área que atiende el caso	Cadena de texto	Operaciones, Atención Clientes, Ventas
AreaProactividadId	Si el caso fue proactivo, el área que inicio el registro del caso	Numérico	Todos los valores para este campo son 0
AreaProactividadNam e	Si el caso fue proactivo, el área que	Cadena de texto	Este campo no contiene valores

	inició el registro del caso		
SortOrder	Campo auxiliar para indicar el orden de registro de casos similares del mismo cliente. No usado en la práctica	Numérico	Todos los valores para este campo son 0
TipoId	El identificador del tipo de incidente o solicitud de servicio del caso	Numérico	Todos los valores para este campo son 0
TipoName	El nombre del tipo de inicidente o solicitud de servicio del caso	Cadena de texto	Falla parcial, Actualizaciones
ClaseId	El identificador de la clase a la que pertenece el caso	Numérico	2, 3
ClaseName1	El nombre de la clase a la que pertenece el caso	Cadena de texto	Solicitud de servicio, Incidente
ClaseName2	El nombre de la clase a la que pertenece el caso. Mismo valor que el campo anterior para los casos de noviembre de 2011 a la fecha	Cadena de texto	Solicitud de servicio, Incidente
PrioridadId	El identificador de la prioridad del caso	Numérico	1, 2, 3
PrioridadName	El nombre del grado de prioridad del caso	Cadena de texto	1 (Bajo), 5 (Alto)
Notificado	Indica si se notificó al cliente del problema detectado o el servicio ofertado si es un caso atendido proactivamente	Numérico	1 (Sí), 0 (No)
NotificaFecha	En caso de que se haya notificado al cliente, la fecha en que se realizó dicha accción	Cadena de texto con formato AAAA-MM-DD hh:mm:ss	2011-11-15 00:56:12
Score	El grado de satisfacción del cliente después de la transacción	Numérico	-1 para indicar que el cliente estuvo insatisfecho con el servicio, 1 para indicar que estuvo

			satisfecho.

1.2 JUSTIFICACIÓN

Mexis cuenta con sistemas que le permiten dar seguimiento en tiempo real la evolución de un caso de atención. De esta manera, es posible conocer en cualquier de manera precisa cuántos casos se están atendiendo en cierto momento, y para cada caso, conocer qué área lo atiende, qué tipo de falla o solicitud de servicio es, cuánto tiempo ha pasado desde que se ingresó al sistema, si fue registrado de modo reactivo o proactivo, y más indicadores que son de utilidad para el negocio. Sin embargo, no es posible conocer el grado de satisfacción que ha experimentado un cliente con un caso en particular sino hasta que el caso es cerrado y el cliente responde la encuesta de satisfacción, por lo que no es posible tomar acción que prevenga que un caso concluya como una experiencia no satisfactoria para el cliente.

Por este motivo es necesario construir un modelo que sea capaz de clasificar los casos tan pronto se registren en el sistema de seguimiento, de tal manera que si un caso se predice como no satisfactorio, el persona de Mexis pueda adoptar medidas especiales para modificar su tendencia, y de ser posible, concluir el caso de manera satisfactoria para el cliente.

1.3 HIPÓTESIS

Mediante el uso de técnicas de minería de datos, es posible construir un modelo clasificador que permita conocer con un buen grado de exactitud, mayor en al menos 10 puntos a los resultados del clasificador Zero R, el grado de satisfacción que tendrá una transacción en un centro de atención telefónica.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERALES

El objetivo principal de este trabajo es generar un árbol de decisión que pueda predecir con un buen grado de precisión el grado de satisfacción que tendrá un cliente después de realizar una transacción en un centro de atención telefónica.

1.4.2 OBJETIVOS PARTICULARES

Los objetivos particulares de este trabajo son:

- Acceder a los datos provistos por Mexis para su transformación.
- Seleccionar los atributos adecuados para la generación de los modelos de clasificación.
- Transformar los atributos seleccionados, en caso de ser necesario, para la generación de los modelos de clasificación.
- Obtener un conjunto de datos listo para ser procesado con los algoritmos Zero R y J48.

- Obtener la taza de clasificación que el algoritmo de clase dominante lograr en el conjunto de datos.
- Realizar experimentos para generar diversos modelos de clasificación.
- Seleccionar el modelo de clasificación con mejor índice de clasificación.
- Obtener reglas representativas a partir del mejor árbol de decisión construido.

1.5 ALCANCES DE LA INVESTIGACIÓN

Mexis cuenta con una amplia cartera de clientes, divididos en dos grupos principales:

- Aliados estratégicos: Este grupo está integrado por aquellos clientes que no contrataron los servicios de Mexis de manera directa, sino que tienen una relación indirecta, teniendo como intermediario a empresas que tiene una relación directa con Mexis, especialmente proveedores de internet mediante banda ancha o enlaces dedicados (ISP). Estos cliente son generalmente consumidores domésticos a los que Mexis brinda soporte en operaciones.

- Clientes de venta directa: Este grupo está integrado por clientes que tienen una relación contractual directa con Mexis. Estos clientes generalmente contratan servicios de seguridad informática, monitoreo de redes, servicios de correo electrónico y hospedaje de páginas web.

Adicionalmente, de acuerdo a la póliza de servicios de los clientes, estos son divididos en cinco segmentos. Cada uno de estos segmentos tiene un tiempo máximo de atención a solicitudes de servicio establecido en su respectivo SLA:

- Diamante y Platino: Tiempo de atención de 2 horas.

- Oro: Tiempo de atención de 3 horas.

- Plata y Bronce: Tiempo de atención de 4 horas.

De acuerdo a expertos de la empresa, los clientes más rentables son los que se encuentran en los segmentos Diamante y Platino. Adicionalmente, la permanencia de un cliente del grupo de Aliados estratégicos obedece a razones de índole económica más que al grado de satisfacción individual de cada cliente, en otras palabras, un cliente de este grupo mantendrá una relación con Mexis a menos que el intermediario terminé su relación con Mexis. Por su parte, la retención de los clientes de venta directa depende en gran medida de su grado de satisfacción, por lo que la calidad de la atención que reciben es más decisiva en su intención de abandono que en el caso de los miembros del primer grupo.

Únicamente se usarán los registros cuyo grupo de área sea Operaciones, ya que se cuentan en el mismo repositorio con registros atendidos por áreas de ventas que siguen un proceso de atención diferente al de los incidentes y solicitudes de servicio.

Para el presente trabajo, se generarán una serie de modelos de clasificación para predecir el grado de satisfacción de casos de solicitud de servicio para clientes que pertenecen a los segmento Diamante y Platino usando el algoritmo de construcción de árboles de decisión C4.5, implementado por la herramienta Weka bajo el nombre de J48, y se seleccionará el modelo que tenga el mejor balance entre precisión y complejidad de la solución.

El proceso de construcción del clasificador es fácilmente replicable para generar un clasificador para los casos de atención de clase “Incidente”.

1.6 RESULTADOS ESPERADOS

Se pretende conseguir un modelo que tenga una tasa de clasificación superior que el clasificador base Zero R. Se espera que el clasificador construido sirva también para obtener reglas que ayuden a mejorar el desempeño del personal del centro de servicio.

2 NATURALEZA DE LOS ALGORITMOS DE MINERÍA DE DATOS Y SU IMPACTO EN LAS APLICACIONES

2.1 MINERÍA DE DATOS Y ALGORITMOS DE MINERÍA DE DATOS

2.1.1 DESCUBRIMIENTO DE CONOCIMIENTO

El descubrimiento de conocimiento, es un proceso automatizado que permite identificar patrones útiles y novedosos en repositorios con grandes cantidades de datos. El núcleo de este proceso se denomina minería de datos. La minería de datos tiene como objetivo construir algoritmos que permitan procesar datos, desarrollar modelos que describan esos datos y encontrar patrones en los mismos (Maimon and Rokach n.d.).

La siguiente figura muestra las fases que componen el proceso de descubrimiento de conocimiento:

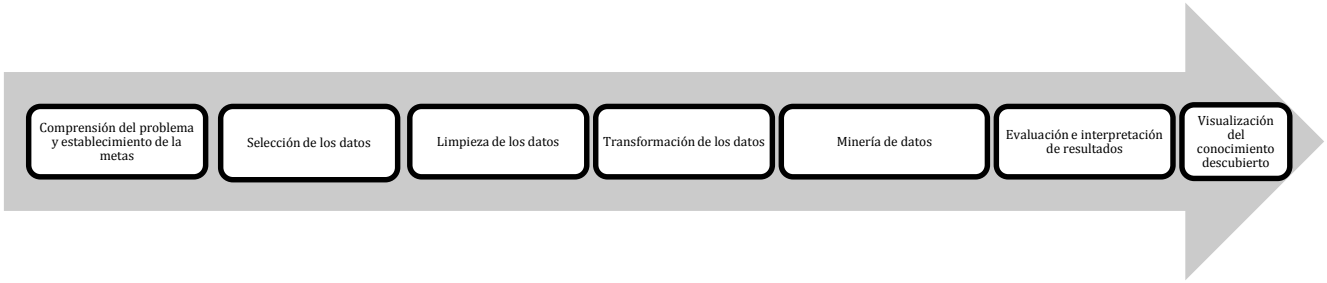


Figura 2.1: Proceso de descubrimiento de conocimiento

Debido a que el descubrimiento del conocimiento es un proceso iterativo, la fase de visualización de los resultados está conectada con el la fase de compresión del problema, de esta manera, el proceso de descubrimiento del conocimiento es continuo, ya que el conocimiento descubierto en iteraciones previas del proceso puede ser usado en iteraciones posteriores, permitiendo nuevos descubrimientos.

A continuación se proporciona una descripción breve de las fases que componen el proceso de descubrimiento de conocimiento:

1. Comprensión del problema y establecimiento de metas: El objetivo de esta etapa es obtener un conocimiento sólido del contexto en que se desarrollará el resto del proceso, así como definir los objetivos del proceso.

2. Selección de datos: En esta etapa se determinan qué datos son necesarios para lograr los objetivos del proceso. Las tareas principales son verificar qué datos están disponibles,

obtener nuevos datos en caso de ser necesario e integrar las fuentes de información seleccionadas en un repositorio único.

3. Preprocesamiento y limpieza de datos: En esta etapa, la información recolectada en la etapa previa es sujeta a procesos que mejoren su calidad, incluyendo actividades como remoción de ruido y valores aislados, y el manejo de valores no presentes.

4. Transformación de datos: El objetivo de esta etapa en generar datos más útiles a la fase posterior. Incluye tareas como la reducción de la dimensionalidad del conjunto de datos y transformaciones en los atributos.

5. Minería de datos: Es la etapa central del proceso, en la que se seleccionan y aplican diferentes algoritmos de acuerdo al tipo de tarea que se desea realizar (clasificación, regresión, agrupamiento).

6. Evaluación e interpretación de resultados: En esta etapa se evalúan e interpretan los resultados obtenidos de la fase previa, en términos de usabilidad, utilidad y comprensibilidad para el usuario.

7. Visualización de resultados o conocimiento obtenido: El objetivo de esta etapa es presentar los resultados obtenidos e integrar el conocimiento obtenido en el proceso de negocios que lo originó, de tal manera que pueda ser aprovechado en condiciones reales.

2.1.2 OBJETIVOS DE LA MINERÍA DE DATOS

Existen dos enfoques básicos de la minería de datos:

- Minería de datos enfocada a la generación de conocimiento (patrones, reglas, etc), también llamada orientada al descubrimiento.
- Minería de datos enfocada a la validación de hipótesis, también llamada orientada a la verificación.

Dentro del proceso de descubrimiento del conocimiento, se usa el primer enfoque. A su vez, este enfoque considera dos objetivos:

- Descripción: Tiene como propósito describir las relaciones en el conjunto de datos , enfocándose en la interpretación y visualización de los datos.
- Predicción: Tiene como propósito construir modelos que permitan aprender el comportamiento de un sistema y ante de la presencia de nuevas muestras (entradas), predecir el valor de una o varias salidas.

En el objetivo predictivo, se cuentan con dos variantes:

- Regresión: Enfocada en la predicción de salidas continuas (i.e: números reales).
- Clasificación: Enfocada en la predicción de salidas discretas (e. g: clases, valores binarios)

Dada la estrecha relación entre la minería de datos y la teoría de aprendizaje – máquina, los métodos de predicción también son conocidos cómo métodos de aprendizaje supervisado, ya que se enfocan en descubrir la relación existente entre un conjunto de entradas, también llamadas

variables independientes y una salida, también llamada variable dependiente. Mediante aprendizaje inductivo, a partir de un conjunto de pares entradas-salida, se consigue generalizar y obtener una relación entre los valores de las entradas y la salida. La relación encontrada es una función o modelo que permite introducir nuevos valores en las entradas y obtener una salida, a pesar de que no exista un registro de entrenamiento con esa combinación de entradas.

La siguiente figura muestra la jerarquía descrita previamente:

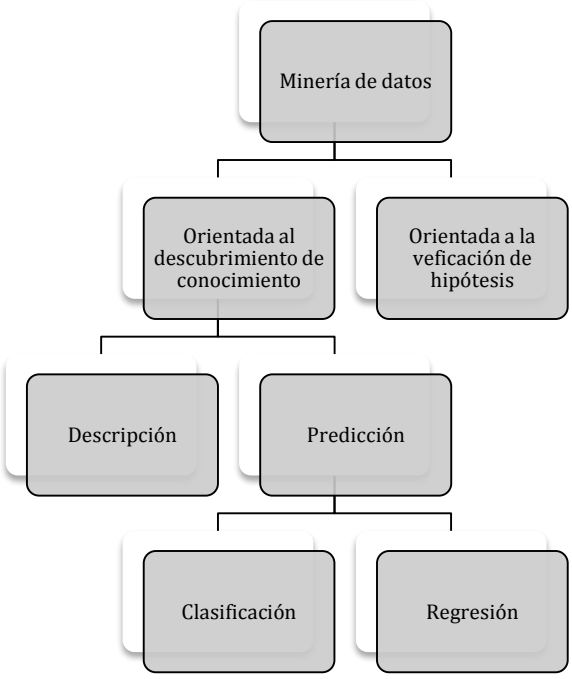


Figura 2.2: Taxonomía de la minería de datos

A continuación se mencionan y describen brevemente algunos de los algoritmos de minería de datos más usados para las tareas de clasificación y regresión.

2.1.3 ALGORITMOS DE MINERÍA DE DATOS

2.1.3.1 Métodos de aprendizaje no supervisado

Los métodos de aprendizaje no supervisado tienen como objetivo modelar la distribución de instancias o registros en un espacio, generalmente de alta dimensionalidad. Dentro del proceso de aprendizaje supervisado, las muestras de entrenamiento no incluyen una variable de salida, sólo contienen entradas. Los algoritmos no supervisados tienen como objetivo construir un modelo que describa estás entradas y evaluar dicho modelo para obtener su exactitud (Kantardzic 2002).

Entre las técnicas más usadas en el aprendizaje no supervisado se encuentran:

- Métodos de visualización.

- Métodos de agrupamiento.
- Reglas de asociación
- Análisis de enlaces.

Los métodos de agrupamiento son probablemente las técnicas de aprendizaje no supervisado más usados. La meta de estos algoritmos es descubrir un conjunto de categorías, útiles e interesantes por su composición, y asignar las muestras proporcionadas a cada una de estas categorías, de tal manera que las muestras similares pertenezcan al mismo grupo. Para lograr esta meta es necesario construir funciones de similaridad o de distancia, donde dos muestras con una distancia entre sí pequeña son muy similares, y dos muestras con una distancia entre sí grande tienen poca similaridad.

Un algoritmo de agrupamiento busca disminuir la distancia entre las muestras que pertenecen a un mismo grupo o categoría, al mismo tiempo que busca maximizar la distancia existente entre cada una de esas categorías.

Entre las medidas de distancia / similaridad más usadas en los algoritmos de agrupamiento se encuentran la métrica Manhattan, la métrica Chebychev y la métrica Minkowski.

2.1.3.2 Métodos de aprendizaje supervisado

Los métodos de aprendizaje supervisado tiene como objetivo principal descubrir la relación existente entre un conjunto de variables independientes y una variable dependiente, también llamada salida. De acuerdo al tipo de la variable objetivo o salida, el aprendizaje supervisado se divide en regresión, donde se busca una salida continua, y clasificación, donde se busca una salida discreta (Hand, Mannila and Smyth 2011).

El proceso de aprendizaje supervisado contempla la existencia de un conjunto de muestras llamada conjunto de entrenamiento que está formado por pares de vectores de entradas – valor de salida. Con este conjunto de aprendizaje el algoritmo construirá un modelo que describa la relación entre las entradas y salidas que se proporcionan.

Para la evaluación del modelo generado con el proceso de aprendizaje supervisado existen diversas técnicas, siendo las más comunes contar con un conjunto de prueba, validar con el mismo conjunto de entrenamiento y usar validación cruzada.

2.1.3.2.1 Algoritmos de regresión

Los algoritmos de regresión son usados para predecir salidas numéricas, por ejemplo:

- La cantidad de visitantes que tendrá una tienda considerando los meses anteriores.
- La cantidad de dinero que puede ganar una empresa en un periodo de tiempo con la venta de cierto producto.

Algunos ejemplos de algoritmos de regresión son:

- Regresión lineal
- Regresión logística.
- Redes neuronales.
- Árboles de regresión.

2.1.3.2.2 Algoritmos de clasificación.

Los algoritmos de clasificación tienen como objetivo construir modelos que predigan salidas de tipo discreto o categórico, por ejemplo, valores binarios (1 / 0 , verdadero / falso, encendido / apagado), o salidas que pertenezcan a clases, por ejemplo, si los valores de la salida pueden estar definidos por la clase colores que contiene los elementos azul, verde, rojo y naranja, y el clasificador indicará qué valor de salida es generado con cierta combinación de entradas.

Algunas de las aplicaciones de los algoritmos de clasificación son:

- Indicar si una persona es susceptible a sufrir un ataque cardiaco.
- Indicar si un solicitante de crédito es sujeto de crédito.
- Indicar a qué especie pertenece una planta o animal usando sus características visibles.

Algunos ejemplos de algoritmos de clasificación son:

- Redes neuronales.
- Máquinas de soporte vectorial.
- Árboles de decisión.
- Redes bayesianas.
- Discriminantes lineales.
- Método del vecino más cercano.

El presente trabajo se centra en el uso de árboles de decisión para la construcción de modelos de predicción, por lo que estos se abordarán con mayor detalle en la siguiente sección.

2.1.3 ÁRBOLES DE DECISIÓN

Los árboles de decisión son uno de los algoritmos de clasificación más populares en las áreas de minería de datos y aprendizaje de máquina. Un árbol de decisión representa un clasificador como una partición recursiva del espacio de instancias. La representación gráfica de un árbol de decisión es un grafo dirigido, con un nodo raíz, es decir, un nodo de inicio que no cuenta con entradas, y con un conjunto de nodos no raíces, cada uno de los cuales cuenta con exactamente una entrada. Los nodos no raíces que cuentan con salidas son llamados internos y los nodos no raíces que no tienen salidas se conocen como terminales u hojas. La siguiente figura provee una representación de un árbol de decisión.

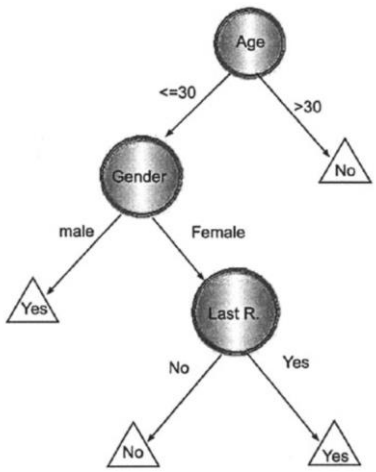


Figura 2.3. Ejemplo de un árbol de decisión

El ejemplo muestra un árbol con el nodo raíz, dos nodos internos y tres nodos terminales.

El nodo raíz y los nodos internos se encargan de dividir el espacio de instancias usando una función sobre los valores de los atributos de entrada. Generalmente, cada nodo interno calcula dicha función sobre un solo atributo del conjunto de entrada. En el caso de atributos discretos, las particiones del espacio de búsqueda se realizan sobre un valor del atributo; en el caso de atributos continuos, las particiones se realizan sobre un rango de valores.

Los inductores de árboles de decisión son algoritmos que construyen árboles de decisión a partir de un conjunto de entrenamiento. Usualmente la construcción del árbol tiene como meta encontrar el clasificador con el menor índice de error de generalización, es decir, el árbol con mayor precisión al clasificar nuevas instancias. Sin embargo, pueden elegirse otros atributos que como objetivo de optimización, como el número de nodos del árbol o la profundidad promedio de las ramas. La complejidad del árbol está en función del número de nodos de contiene, y generalmente se busca que un algoritmo inductor genere árboles de poca complejidad ya que son más fáciles de interpretar por una persona.

El problema de encontrar una árbol de decisión óptimo, es decir, que minimice el error de generalización para un conjunto de entrenamiento dado, es considerado complejo. De acuerdo a un trabajo de Hancok [10] encontrar el árbol de decisión mínimo para un conjunto de entrenamiento proporcionado es un problema NP duro. Por lo tanto, es necesario usar heurísticas que permitan encontrar árboles con un buen grado de precisión, sin necesariamente ser óptimos. Existen diversos algoritmos inductores que permiten generar árboles con un enfoque descendente (*top-down*) [10], como ID3, propuesto por Quinlan[11], C4.5, también propuesto por Quinlan[12] y CART, creado por Breiman [13]. Estos algoritmos constan generalmente de dos fases, la primera donde el árbol de decisión es generado usando las instancias de entrenamiento proporcionadas, también llamada “crecimiento” y una fase donde el árbol es reducido de tamaño para mejorar su capacidad de generalización, disminuyendo el sobreentrenamiento [14], también llamada “poda”. Algunos algoritmos únicamente implementan la primera fase, sin realizar poda sobre el árbol construido.

2.1.4 ALGORITMO C4.5

El algoritmo C4.5 propuesto por Quinlann [12] es una evolución de su algoritmo previo ID3. A diferencia de su predecesor, C4.5 tiene la capacidad de manejar atributos continuos mediante la construcción de umbrales (ID3 únicamente puede trabajar con atributos discretos), puede manejar valores nulos o faltantes en los atributos y permite ajustar el grado de la poda que se realiza posterior a la construcción del árbol.

2.1.5 IMPLEMENTACIÓN J48

El algoritmo C4.5 cuenta con diferentes implementaciones, pero probablemente la más conocida es el clasificador J48 incluido en la herramienta Weka (The University of Waikato n.d.).

La implementación J48 contiene varios parámetros de ejecución que permiten controlar la forma en que se generan los árboles. Los parámetros más relevantes son:

- *BinarySplit*. Si el valor de este atributo es verdadero, al momento de generar el árbol de decisión, los nodos internos únicamente pueden tener dos salidas. Si el valor es falso, un nodo interno puede tener más de dos salidas, siendo el número máximo de salidas la cardinalidad del conjunto de valores del atributo que se está usando para realizar el corte en dicho nodo.

- *ConfidenceFactor*: El factor de confianza determina el grado de poda que se realizará posterior a la creación del árbol, a mayor valor del factor de confianza, menor es la cantidad de poda que se realiza en el árbol. Los valores usados están entre 0 y 0.5. Si el valor se establece en 0,

- *Debug*: Parámetro de ejecución que determina la cantidad de información que se despliega en pantalla durante la ejecución.

- *MinNumObj*: Indica el número mínimo de objetos que debe contener un nodo para conservarse, siendo relevante para la poda del árbol durante su creación (Drazin and Montag n.d.).

- *SaveInstanceData*: Parámetro que indica si se guardarán las instancias de entrenamiento para desplegarse junto con el árbol de decisión.

- *Unpruned*: Si el valor del parámetro es verdadero, no se realizará poda del árbol, si el valor es falso el árbol será podado durante y posteriormente a la ejecución.

Los parámetros de ejecución que tienen una influencia directa en la complejidad del árbol de decisión y por lo tanto en su precisión son *ConfidenceFactor*, *MinNumObj* y *BinarySplit.*

2.2 ESTADO DEL ARTE

Existe una diversidad de trabajos relacionados en los que me he basado para la realización de este trabajo. En 2002, Garver [12] recopiló una serie de aplicaciones de diversos algoritmos de minería de datos enfocados a la investigación sobre satisfacción del cliente, desde un enfoque de satisfacción acumulada. Dentro de ese trabajo propone el uso de redes neuronales para la evaluación de los atributos que impactan en el grado de satisfacción de un cliente y cómo la satisfacción del cliente afecta el rendimiento financiero de la empresa con la que tiene relación. Para su trabajo, Garver usó datos de una cadena de pizzas para conocer las percepciones del cliente con respecto a los atributos de la pizza, como el sabor, la temperatura, el tamaño, precio; y atributos del servicio de entrega, como rapidez de la entrega, amabilidad del repartidor y confiabilidad de la entrega; y qué tan satisfecho está el producto. Se pidió que las personas evaluaran cada uno de estos factores en una escala numérica de 1 a 10. Posteriormente se usaron redes neuronales, colocando cada uno de estos factores como una neurona en la capa de entrada de la red y usando análisis de sensibilidad para determinar qué importancia tiene cada factor en el cálculo de la satisfacción global de cada cliente. De esta manera, se determinó que el sabor, el precio y la confiabilidad de la entrega son los atributos más importantes para predecir el grado de satisfacción del cliente con el producto y por ende, con la cadena de pizzas. Dentro del mismo trabajo se utilizaron árboles de decisión para predecir la lealtad del cliente, usando como variables independientes los atributos del producto y del servicio de entrega; como resultado se obtuvo que los clientes leales dan calificaciones altas a los atributos precio y confiabilidad de la entrega, mientras que los clientes no leales dan calificaciones bajas a los atributos sabor y tamaño del producto.

Park y Gates [16] presentaron en 2009 un trabajo que tiene como objetivo predecir la satisfacción transaccional de clientes de una mesa de servicio mediante el uso de técnicas de transcripción automática de llamadas en tiempo real, y minería de datos, en su variable de minería de texto sobre dichas transcripciones, además del uso de algoritmos tradicionales como Máquinas de soporte de vectores, Naive Bayes y Regresión logística. Se realizaron diversos experimentos sobre la información de llamadas al centro de atención telefónica de una compañía de automóviles. Se usaron dos escalas para la satisfacción de cada transacción, una de dos valores y una de cinco valores, aplicando los algoritmos en ambos casos y comparando los resultados contra la salida de un algoritmo de clase dominante, similar al algoritmo Zero R utilizado en este trabajo y contra las predicciones de un experto del negocio. Se encontró que usando algoritmos de minería de datos es posible construir clasificadores que tienen mejor porcentaje de clasificación que los clasificadores de clase dominante e incluso que las predicciones del experto, alcanzando 89.42% de exactitud en el caso de la escala de dos valores para la satisfacción, y 66.09% en el caso de la escala de cinco valores.

3 PROCESAMIENTO PREVIO A LA GENERACIÓN DE LOS CLASIFICADORES

3.1 OBTENCIÓN DE DATOS

3.2 SELECCIÓN DE DATOS

3.3 PREPROCESAMIENTO Y LIMPIEZA DE DATOS

3.4 TRANSFORMACIÓN DE DATOS

ETAPA 1: RECOLECCIÓN DE DATOS.

ETAPA 2: PREPROCESAMIENTO DE LOS DATOS.

Selección de datos basada en el Segmento de atención del cliente.

Objetivo: Limitar el número de datos con el que se trabajará.

Procedimiento: El archivo resultante de la consulta a la tabla *TicketDimensions* de la base de datos de SQL Server proporcionados por Mexis, denominado *TicketDimensions.csv*, que contiene 121, 461 registros, se modificará en MS Excel.

- Aplicar un filtro sobre la columna *CustomerSegmentoName*, seleccionado únicamente los valores *Diamante* y *Platino.* Se obtiene un archivo denominado *TicketDimensions1.csv* que contiene 51, 844 registros.

Selección de datos basada en el Canal de Marketing del cliente.

Objetivo: Limitar el número de datos con el que se trabajará.

Procedimiento: El archivo *TicketDimensions1.csv* resultado del paso anterior, se modificará en MS Excel.

- Aplicar un filtro sobre la columna *CanalMarketingName*, seleccionado únicamente el valor *Venta directa.* Se obtiene un archivo denominado *TicketDimensions2.csv* que contiene 25, 464 registros.

Selección de datos basada en el Grupo de atención del caso.

Objetivo: Limitar el número de datos con el que se trabajará.

Procedimiento: El archivo *TicketDimensions2.csv* resultado del paso anterior, se modificará en MS Excel.

- Aplicar un filtro sobre la columna *GrupoAreaName2*, seleccionado únicamente el valor *OPERACIONES.* Se obtiene un archivo denominado *TicketDimensions3.csv* que contiene 23, 202 registros.

División de datos basada en la clase del ticket.

Objetivo: Crear dos conjuntos de datos que entrenarán modelos de comportamientos diferentes.

Procedimiento: El archivo *TicketDimensions3.csv* resultado del paso anterior se modificará en MS Excel.

- Aplicar un filtro sobre la columna *ClaseName2,* seleccionado únicamente el valor *FALLA.* Se creará un archivo denominado *Datos Incidentes.csv* para almacenar los registros seleccionados. El archivo contiene 10, 986 registros.
- Aplicar un filtro sobre la columna *ClaseName2,* seleccionado únicamente el valor *GESTIÓN.* Se creará un archivo denominado *Datos Solicitudes servicio.csv* para almacenar los registros seleccionados. El archivo contiene 9, 647 registros.

Limpieza de los conjuntos de datos.
Objetivo: Asegurar que los valores que contienen los conjuntos de datos son validos.
Procedimiento: Cada uno de los archivos generados en el paso anterior (*Datos Incidentes.csv* y *Datos Solicitudes servicio.csv*), se modificará en MS Excel aplicando los siguientes pasos.

- Aplicar un filtro sobre la columna *TiempoTotal,* eligiendo solamente los registros con valores positivos.
- Aplicar un filtro sobre la columna *CreatedStatffId,* eligiendo únicamente los registros con valores mayores a 1.

Se crearán dos archivos llamados *Datos Incidentes 1.csv,* con 10, 640 registros, que almacenará los registros procesados de *Datos Incidentes.csv,* y *Datos Solicitudes servicio 1.csv,* con 6, 763 registros que almacenará los registros procesados de *Datos Solicitudes servicio.csv.*
Selección de atributos que a utilizar.
Objetivo: Disminuir la dimensionalidad del conjunto de datos con el que se trabajará y que será almacenado en la base de datos propia del modelo, mediante la eliminación de características irrelevantes, duplicadas o que únicamente agreguen ruido al modelo de datos.
Procedimiento: Se definirá un proceso separado para cada uno de los archivos generados en el paso anterior.
Para el archivo *Datos Incidentes 1.csv* se realizarán las siguientes actividades usando MS Excel:

- El atributo *TicketId* se conserva sin modificación como referencia para posibles consultas futuras.
- El atributo *ProjectId* se eliminará porque no aporta información al modelo, dado que únicamente toma el valor *0* para todos los casos.
- El atributo *EnteredPeriod* se eliminará ya que su información está contenida en el campo *EnteredDateTime.*
- El atributo *EnteredDate* se eliminará ya que su información está contenida en el campo *EnteredDateTime.*
- Se creará un atributo llamado *Year*, que será el resultado de aplicar la función *AÑO()* al campo *EnteredDateTime.*
- Se creará un atributo llamado *Month*, que será el resultado de aplicar la función *MES()* al campo *EnteredDateTime.*
- Se creará un atributo llamado *Day*, que será el resultado de aplicar la función *DIA()* al campo *EnteredDateTime.*
- Se creará un atributo llamado *DayName,* que será el resultado de aplicar la función *DIASEM()* al campo *EnteredDateTime.*
- Se creará un atributo llamado *Hour*, que será el resultado de aplicar la función *HORA()* al campo *EnteredDateTime.*

- Se creará un atributo llamado *Minute*, que será el resultado de aplicar la función *MINUTO()* al campo *EnteredDateTime.*
- Se removerá el atributo *EnteredDateTime* ya que sus valores quedaron definidos por los campos que se crearon anteriormente (*Year, Month, Day, Hour, Minute*).
- El atributo *EnteredWeekNameId* se conservará sin modificación ya que permitirá la comprobación sencilla de casos similares que se presentan en la misma semana.
- El atributo *EnteredWeekName* se eliminará ya que puede ser calculado usando el atributo *EnteredWeekNameId* y un calendario apropiado.
- El atributo *CustomerId* se conservará sin modificación como referencia para posible consultas futuras.
- El atributo *CustomerName* se eliminará eligiendo en su lugar el campo *CustomerShortName* ya que este es de menor longitud y conserva la cantidad de información necesaria.
- El atributo *CustomerActive* ser eliminará ya que la mayoría de los casos fueron levantados por clientes que se encontraban activos al momento de levantar el ticket, además de que no muestra el valor del atributo al momento del registro del ticket, sino al momento de la consulta de los mismos, impidiendo que se conozca el momento exacto en que se dieron los cambios de estado del cliente.
- El atributo *CustomerStaffId* se eliminará ya que no proporcionará información constante al modelo debido a la constante rotación de personal.
- El atributo *CustomerStaffName* se eliminará por la misma razón del campo anterior, debido a que existe una relación 1 a 1 entre ambos.
- El atributo *CustomerSalesId* se eliminará ya que no proporciona información relevante al modelo.
- El atributo *CustomerSalesName* se eliminará por la misma razón del atributo anterior, debido a que existe una relación 1 a 1 entre ambos.
- El atributo *CustomerSegmentoId* se eliminará eligiendo en su lugar al campo *CustomerSegmentoName,* siendo el segundo auto descriptivo con relación a primero.
- El atributo *CustomerSegmentoName* se conservará ya que contiene 2 valores que pueden ser representativos para el modelo, pero su nombre será cambiado a *CustomerSegmentName*.
- El atributo *CustomerSegmentoAtencionName1* se eliminará ya que tiene un único valor para todos los registros (ODyP).
- El atributo *CustomerRegionId* se eliminará eligiendo en su lugar al campo *CustomerRegionName,* siendo el segundo auto descriptivo con relación a primero.
- El atributo *CustomerRegionName* se conservará sin modificaciones.
- El atributo *AssignedStaffId* se eliminará ya que no proporciona información constante al modelo debido a la rotación de personal de atención.
- El atributo *AssignedStaffName* se eliminará por la misma razón que el campo anterior, debido a que existe una relación 1 a 1 entre ambos.

- El atributo *AssignedDireccionId* se eliminará eligiendo en su lugar al campo *AssignedDireccionName,* siendo el segundo auto descriptivo con relación al primero.
- El atributo *AssignedDireccionName* se conservará sin modificaciones en sus valores, pero su nombre será cambiado a *AssignedSeniorManagementName* por razones de estandarización.
- El atributo *AssignedGerenciaId* se eliminará eligiendo en su lugar al campo *AssignedGerenciaName,* siendo el segundo auto descriptivo con relación al primero.
- El atributo *AssignedGerenciaName* se conservará sin modificaciones en sus valores, pero su nombre será cambiado a *AssignedManagementName* por razones de estandarización.
- El atributo *CreatedStaffId* se eliminará debido a que no aporta información constante al modelo a causa de la rotación de personal de atención.
- El atributo *CreatedStaffName* se eliminará por la misma razón que el campo anterior, debido a que existe una relación 1 a 1 entre ambos.
- El atributo *CreatedDireccionId* se eliminará eligiendo en su lugar al campo *CreatedDireccionName,* siendo el segundo auto descriptivo con relación al primero.
- El atributo *CreatedDireccionName* se conservará sin modificación en sus valores, pero su nombre será cambiado a *CreatedSeniorManagementName* por razones de estandarización.
- El atributo *CreatedGerenciaId* se eliminará eligiendo en su lugar al campo *CreatedGerenciaName,* siendo el segundo auto descriptivo con relación al primero.
- El atributo *CreatedGerenciaName* se conservará sin modificaciones en sus valores, pero su nombre será cambiado a *CreatedManagementName* por razones de estandarización.
- El atributo *StatusId* se eliminará debido a que no proporciona información relevante al modelo, ya que muestra únicamente el último estado (o estado actual) de los tickets, siendo este en general, *1001* o *Cerrados.*
- El atributo *StatusName* se eliminará por la misma razón que el campo anterior, debido a que existe una relación 1 a 1 entre ambos.
- El atributo *AreaId* se eliminará eligiendo en su lugar al campo *AreaName2*, siendo el segundo auto descriptivo con relación al primero.
- El atributo *AreaName1* se eliminará eligiendo en su lugar al campo *AreaName2*, siendo el segundo más descriptivo que el primero.
- El atributo *AreaName2 se* conservará sin cambio en sus valores, pero su nombre será cambiado a *AreaName* por simplicidad.
- El atributo *CategoriaId* se eliminará eligiendo en su lugar al campo *CategoriaName,* siendo el segundo auto descriptivo con relación al primero.
- El atributo *CategoriaName* se conservará sin cambio en sus valores, pero su nombre será cambiado a *CategoryName* por razones de estandarización.
- El atributo *AccionId* se eliminará eligiendo en su lugar al campo *AccionName,* siendo el segundo auto descriptivo con relación al primero.
- El atributo *AccionName* se conservará sin cambios en sus valores, pero su nombre será cambiado a *ActionName* por razones de estandarización.

- El atributo *GrupoAreaId* se eliminará ya que no aporta información adicional al modelo, teniendo únicamente el valor *2* para todos los registros.
- El atributo *GrupoAreaName1* se eliminará ya que no aporta información adicional al modelo, teniendo únicamente el valor *OPE* para todos los registros.
- El atributo *GrupoAreaName2* se eliminará ya que no aporta información adicional al modelo, teniendo únicamente el valor *OPERACIONES* para todos los registros.
- El atributo *SortOrder* se eliminará ya que no aporta información relevante al modelo, siendo un valor de control y no de operación.
- El atributo *TipoId* se conservará como referencia para posibles consultas futuras, pero su nombre se cambiará a *TypeId* por razones de estandarización.
- El atributo *TipoName* se conservará sin cambio en sus valores, pero su nombre será cambiad a *TypeName* por razones de estandarización.
- El atributo *ClaseId* se eliminará ya que no aporta información adicional al modelo, teniendo únicamente el valor *2* para todos los registros.
- El atributo *ClaseName1* se eliminará ya que no aporta información adicional al modelo, teniendo únicamente el valor *FALLA* para todos los registros.
- El atributo *ClaseName2* se eliminará ya que no aporta información adicional al modelo, teniendo únicamente el valor *FALLA* para todos los registros.
- El atributo *PrioridadId* se eliminará eligiendo en su lugar el campo *PrioridadName,* siendo el segundo auto descriptivo con relación al primero.
- El atributo *PrioridadName* se conservará sin cambio en sus valores, pero su nombre será cambiado a *Priority* por razones de estandarización y simplicidad.
- El atributo *Enviada* se conservará sin cambio en sus valores, pero su nombre será cambiado a *Send* por razones de estandarización.
- El atributo *Insatisfecho* se conservará sin cambio en sus valores, pero su nombre será cambiado a *Dissatisfied* por razones de estandarización.
- El atributo *LastUpdatePeriod* será eliminado ya que no aporte información relevante al modelo, siendo un campo de control más que de operación.
- El atributo ‘*No requerida’* se conservará sin cambio en sus valores, pero su nombre será cambiado a *NotRequired* por razones de estandarización.
- El atributo *No enviada* se eliminará ya que no aporta información adicional al modelo, pudiendo ser calculada usando los campos *NotRequired y Send.*
- El atributo *Satisfecho* se conservará sin cambio en sus valores, pero su nombre será cambiado a *Satisfied* por razones de estandarización.
- El atributo *SentSurveyId* se eliminará ya que no aporta información adicional al modelo, debido a que sus valores pueden ser calculada usando los campos *NotRequired* y *Send.*
- El atributo *SentSurveyName* se eliminará por la misma razón que el atributo anterior, debido a que existe una relación 1 a 1 entre ambos.
- El atributo *RangoTiempoMexis* se eliminará debido a que no aporta información adicional al modelo, ya que sus valores pueden ser calculados usando el atributo *TiempoMexis.*

- El atributo *RangoTiempoMexisHoras* se eliminará debido a que no aporta información adicional al modelo, ya que sus valores pueden ser calculados usando el atributo *TiempoMexis.*
- El atributo *RangoTiempoTotal* se eliminará debido a que no aporta información adicional al modelo, ya que sus valores pueden ser calculados usando el atributo *TiempoTotal.*
- Se creará un atributo llamado *MexisTimeMinutes* cuyos valores serán el resultado de aplicar la formula REDONDEAR.MAS(), al valor correspondiente del campo *TiempoMexis* dividido entre 60 y con 0 como segundo parámetro.
- El atributo *TiempoMexis* será eliminado ya que sus valores pueden ser calculados usando el campo *MexisTimeMinutes*, además de que su nivel de detalle puede agregar ruido al modelo.
- Se creará un atributo llamado *TotalTimeMinutes* cuyos valores serán el resultado de aplicar la formula REDONDEAR.MAS(), al valor correspondiente del campo *TiempoTotal* dividido entre 60 y con 0 como segundo parámetro.
- El atributo *TiempoTotal* será eliminado ya que sus valores pueden ser calculados usando el campo *TotalTimeMinutes*, además de que su nivel de detalle puede agregar ruido al modelo.
- El atributo *AreaProactividadId* será eliminado eligiendo en su lugar al atributo *AreaProactividadName,* siendo el segundo auto descriptivo con relación al primero.
- El atributo *AreaProactividadName* se conservará sin modificación en sus valores, pero su nombre será cambiado a *AreaProactivityName* por razones de estandarización.
- El atributo *GuarantorId* se eliminará eligiendo en su lugar al campo *GuarantorShortName,* siendo el segundo auto descriptivo con relación al primero.
- El atributo *GuarantorName* se eliminará eligiendo en su lugar al campo *GuarantorShortName,* ya que el segundo conserva la misma cantidad de información que el primero pero con una longitud menor.
- El atributo *RangoTiempoMexisSinMonitoreo* se eliminará debido a que no aporta información adicional al modelo, ya que su nivel de detalle es menor que manejado para otros atributos, esto es, este atributo tiene una medición en rango de horas, mientras que los campos de tiempo restantes están medidos en minutos.
- El atributo *RangoTiempoMexisSinMonitoreoHoras* se eliminará debido a que no aporta información adicional al modelo, ya que su nivel de detalle es menor que manejado para otros atributos, esto es, este atributo tiene una medición en horas, mientras que los campos de tiempo restantes están medidos en minutos.
- El atributo *CanalMarketingName* se eliminará debido a que no aporta información adicional al modelo, siendo su único valor para todos los casos ‘Venta directa’.
- El atributo *GuarantorShortName* se conservará sin modificaciones.
- El atributo *SegmentoMercadoName* se eliminará debido a que no aporta información relevante al modelo, ya que sus valores generalmente coinciden con los del atributo *CustomerSegmentName* y en los casos en que no se presenta dicha condición se puede generar ruido a la técnica de clasificación.

- El atributo *EnteredQuarter* se eliminará ya que no aporta información adicional al modelo, debido a que sus valores pueden ser calculados usando los atributos *Year* y *Month.*
- El atributo *Notificado* será conservará sin modificación en sus valores, pero su nombre será cambiado a *Notified* por razones de estandarización.
- El atributo *NotificaFecha* será eliminado ya que no está correctamente estructurado, debido a que no especifica la fecha y hora exacta de notificación, sino que únicamente proporciona una hora y minuto sin indicar a qué fecha pertenecen dichos valores.
- El atributo *NotificaStaff* será eliminado debido a que no proporciona información constante al modelo, a causa de la rotación de personal.
- Se agregará un atributo llamado *ChangeCategoryTimes,* cuyos valores se integrarán posteriormente, colocándolos por el momento en 0.
- Se agregará un atributo llamado *ChangeTypeTimes,* cuyos valores se integrarán posteriormente, colocándolos por el momento en 0.
- Se agregará un atributo llamado *ChangeAssigmentsTimes,* cuyos valores se integrarán posteriormente, colocándolos por el momento en 0.
- Se agregará un atributo llamado *SameTypeAtWeek,* cuyos valores serán calculados posteriormente, colocándolos por el momento en 0.
- Se agregará un atributo llamado *Index* cuyos valores serán calculados posteriormente, colocándolos por el momento en 0.

Se creará un archivo denominado *Datos Incidentes 2.csv* que contiene únicamente los campos especificados anteriormente. Dado que no se agregaron o eliminaron registros, *Datos Incidentes 2.csv* contiene la misma cantidad de filas que *Datos Incidentes 1.csv*, es decir, 10, 640 registros.

La siguiente tabla muestra la lista de atributos que integrarán el archivo *Datos Incidentes 2.csv*

Eliminación de datos duplicados en el conjunto de datos.

Objetivo: Disminuir el tamaño de los conjunto de datos y conservar la integridad en la base de datos mediante el uso de datos no duplicados en los conjuntos de entrenamiento.
Procedimiento: Usando *Pentaho Data Integration* se aplicará una transformación a cada uno de los conjuntos de datos. Para cada uno de los conjuntos, se creará una transformación. Para el conjunto, *Datos Incidentes 2.csv*, la transformación se llamará *Remover Duplicados Incidentes.ktr*, teniendo como salida un archivo denominado *Datos Incidentes 3.csv*, que contiene únicamente los registros no duplicados y una sola copia de los registros duplicados; para el conjunto *Datos Solicitudes servicio 2.csv*, la transformación se llamará *Remover Duplicados Solicitudes Servicio.ktr*, teniendo como salida un archivo llamado *Datos Solicitudes servicio 3.csv*, que contiene únicamente los registros no duplicados y una sola copia de los registros no duplicados.

Transformación *Remover Duplicados Incidentes*

Transformación *Remover Duplicados Incidentes*

Los pasos a seguir son:

1. Leer el archivo de entrada.
2. Ordenar los registros en forma ascendente usando la llave primaria (TicketId)
3. Remover las filas duplicadas.
4. Dirigir la salida a un nuevo archivo *csv*.

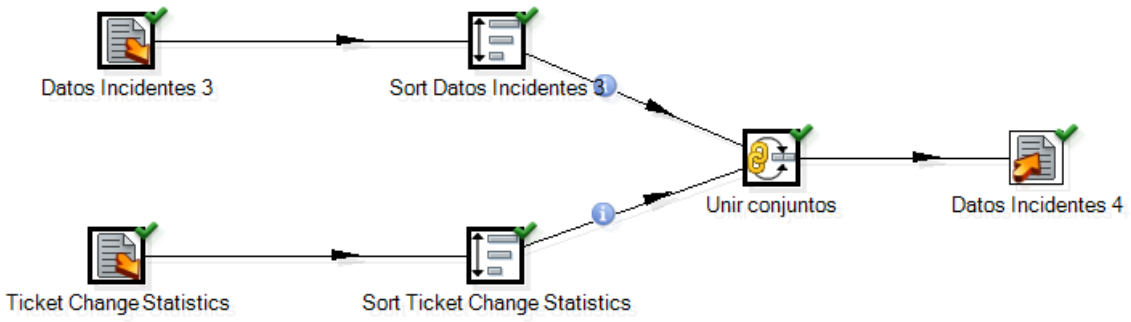
El archivo *Datos Incidentes 3.csv* contiene 10, 599 registros.
El archivo *Datos Solicitudes servicio 3.csv* contiene **Pendientes**
Adición de los valores de los atributos ChangeCategoryTimes, ChangeTypeTimes y ChangeAssigmentTimes al conjunto de datos.

Objetivo: Unir los conjuntos de datos actuales con el archivo *Ticket Change Statistics.csv* con el fin de agregar los valores adecuados de los campos ChangeCategoryTimes, ChangeTypeTimes y ChangeAssigmentTimes.

Procedimiento: Usando Pentaho Data Integration se aplicará una transformación al conjunto de datos *Datos Solicitudes Servicios 3.csv* para unir los atributos de dicho conjunto con los atributos del conjunto *Ticket Change Statistics.csv*. Dicha transformación tendrá el nombre *Unir Incidentes y Ticket Statistics.krt*

Los pasos para realizar dicho proceso son los siguientes:

1) Leer el archivo de entrada *Datos Solicitudes Servicio 3.csv*. Remover los campos *ChangeCategoryTimes*, *ChangeTypeTimes* y *ChangeAssigmentTimes* del paso de lectura de archivo. Aplicar un ordenamiento sobre la salida de lectura del archivo usando como llave el atributo *TicketId.*
2) Leer el archivo de entrada *Ticket Change Statistics.csv*. Aplicar un ordenamiento sobre la salida de lectura del archivo usando como llave el atributo *TicketId*.
3) Agregar un paso *Merge Join* y colocar como entrada las salidas ordenadas de cada archivo, tomando como llave el campo *TicketId* en cada entrada.
4) Redirigir la salida de la unión a un archivo llamado *Datos Incidentes 4.csv,* eliminando el campo *TicketId_1* y colocando el atributo Index en la última posición.



Transformación *Unir Incidentes y Ticket Statistics*

El conjunto *Datos Incidentes 4.csv* contiene 10, 345 registros.
Carga a la base de datos.

Objetivo: Cargar el conjunto de datos *Datos Incidentes 4*.csv a una base de datos para aplicar el procedimiento de cálculo del índice de satisfacción.

Procedimiento: Se creará una transformación en Pentaho denominada *Cargar Datos Incidentes 4.ktr*.

Los pasos a realizar son los siguientes:

1) Leer el archivo *Datos Incidentes 4.csv.*
2) Redirigir la salida a una tabla de la base de datos.

Como requisito previo, es necesario crear una tabla en MySQL, denominada *MPS*, y una tabla llamada *TDFaultDP* con los datos especificados en el diagrama inferior.

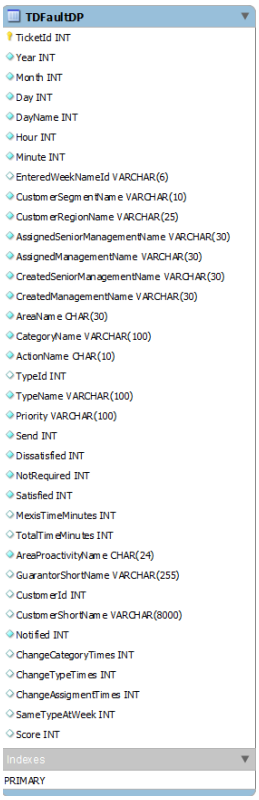


Tabla TDFaultDP

El diagrama completo de la base de datos y los scripts correspondientes se encuentran en la sección Anexos **(Pendiente).**

El segundo paso del proceso necesita de una conexión a la base de datos *MPS.* A dicha conexión se deberán proporcionar los valores por adecuados, es decir, asignar privilegios de escritura. Se utilizará la opción de detección y asociación automática de campos entre el archivo y la tabla *TDFaultDP*, ya que ambos presentan la misma cantidad de campos y tipos de datos similares.

Transformación *Cargar Datos Incidentes 4.*

4 EXPERIMENTOS Y RESULTADOS

4.1 DESCRIPCIÓN DE LOS EXPERIMENTOS

4.1.1 PARÁMETROS DE EJECUCIÓN DEL ALGORITMO J48

Por defecto, Weka ofrece el valor 0.25 para el parámetro factor de confianza y 2 para el número mínimo de objetos por hoja. Para verificar el efecto que tienen estos parámetros en el grado de precisón y la complejidad del árbol de decisión que genera el algoritmo, se realizarán diversos experimentos en los que se ejecutará el algoritmo J48 con diferentes valores en los parámetros señalados previamente, con el objetivo de encontrar el árbol que tenga la mejor precisión para el conjunto de datos obtenido en el capítulo anterior.

Los valores que se probarán para cada parámetro son los siguientes:

- *ConfidenceFactor:* 0.1, 0.2, 0.3, 0.4 y 0.5
- *MinNumObj:* 2, 5, 10

Considerando las combinaciones posibles de estos valores, se generarán 15 árboles de decisión y se elegirá el clasificador con mayor precisión. Si dos clasificadores tienen la misma precisión, se elegirá como mejor aquél cuya matriz de confusión favorezca la predicción de casos insatisfechos.

La lista completa de experimentos se muestra en la siguiente tabla:

Tabla 4.1 Lista de experimentos a realizar

Número de experimento	Factor de confianza	Número mínimo de objetos por nodo
1	0.1	2
2	0.1	5
3	0.1	10
4	0.2	2
5	0.2	5
6	0.2	10
7	0.3	2
8	0.3	5
9	0.3	10
10	0.4	2
11	0.4	5
12	0.4	10
13	0.5	2
14	0.5	5
15	0.5	10

4.1.2 MÉTODO DE VALIDACIÓN DE LOS MODELOS

Dada la pequeña cantidad de instancias de entrenamiento con los que se cuentan, se utilizará validación cruzada de 10 hojas para obtener la precisión de cada árbol generado.

4.1 REALIZACIÓN DE EXPERIMENTOS

En la sección anterior se indicaron los parámetros de ejecución que se usarán para la realización de los experimentos. Esta sección está destinada a ejecutar los experimentos y analizar sus resultados. Como resultado de cada ejecución, se creará un árbol de decisión del que se obtendrá el porcentaje de clasificación correcto de las instancias de entrenamiento para compararlo con el resto de los resultados. Adicionalmente, como se mencionó en los objetivos, los resultados de cada modelo generado se compararán con el resultado de ejecutar el algoritmo ZeroR o de clase dominante sobre el mismo conjunto de datos.

4.1.1 EXPERIMENTO 1

Los resultados del experimento número 1 son los siguientes:

Tabla 4.2 Resultados experimento 1

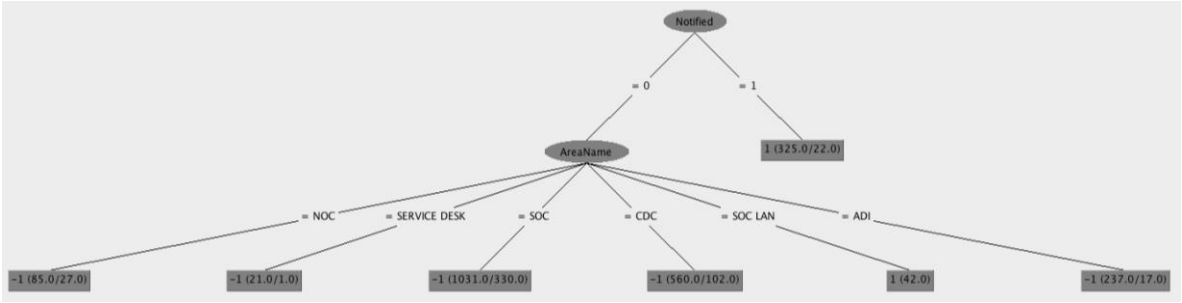
Instancias clasificadas correctamente	1780
Instancias clasificadas incorrectamente	521
Porcentaje de clasificación	77.3577%
Número de nodos	7
Número de hojas (nodos terminales)	9

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.3 Matriz de confusión del experimento 1

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1457	22
Satisfecho	499	323

Figura 4.1 Árbol de decisión generado en el experimento 1



4.1.2 EXPERIMENTO 2

Los resultados del experimento número 2 son los siguientes:

Tabla 4.4 Resultados experimento 2

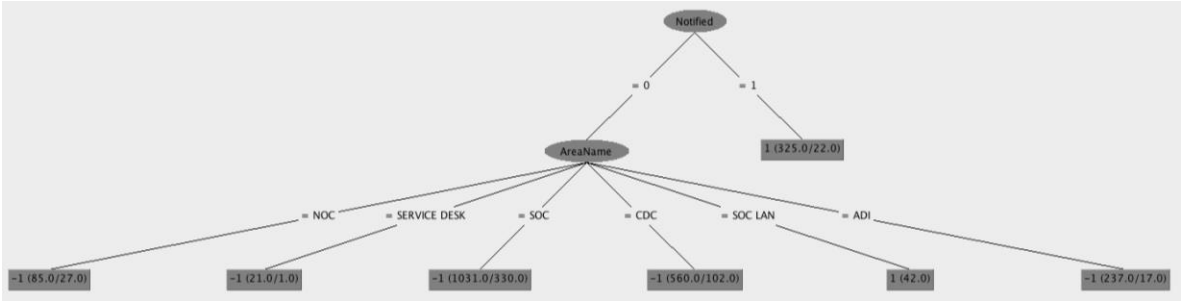
Instancias clasificadas correctamente	1780
Instancias clasificadas incorrectamente	521
Porcentaje de clasificación	77.3577%
Número de nodos	7
Número de hojas (nodos terminales)	9

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.5 Matriz de confusión del experimento 2

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1457	22
Satisfecho	499	323

Figura 4.2 Árbol de decisión generado en el experimento 2



4.1.3 EXPERIMENTO 3

Los resultados del experimento número 3 son los siguientes:

Tabla 4.6 Resultados experimento 3

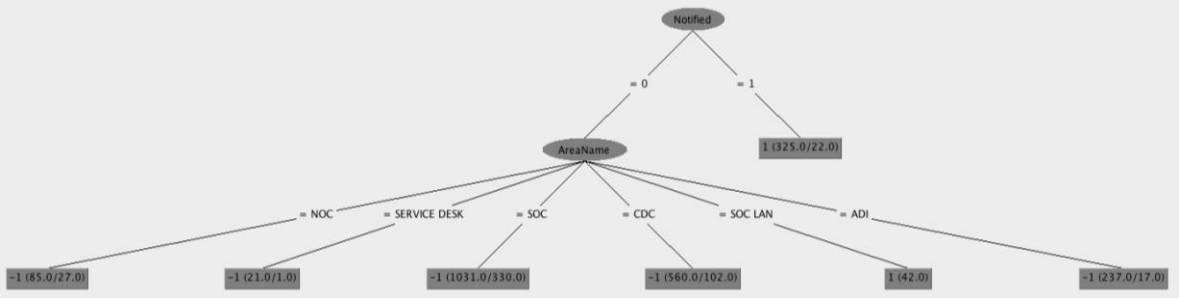
Instancias clasificadas correctamente	1780
Instancias clasificadas incorrectamente	521
Porcentaje de clasificación	77.3577%
Número de nodos	7
Número de hojas (nodos terminales)	9

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.7 Matriz de confusión del experimento 3

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1457	22
Satisfecho	499	323

Figura 4.3 Árbol de decisión generado en el experimento 3



4.1.4 EXPERIMENTO 4

Los resultados del experimento número 4 son los siguientes:

Tabla 4.8 Resultados experimento 4

Instancias clasificadas correctamente	1796
Instancias clasificadas incorrectamente	505
Porcentaje de clasificación	78.053%
Número de nodos	35
Número de hojas (nodos terminales)	39

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.9 Matriz de confusión del experimento 4

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1445	34
Satisfecho	471	351

La representación gráfica del árbol es muy por lo que se omite.

4.1.5 EXPERIMENTO 5

Los resultados del experimento número 5 son los siguientes:

Tabla 4.10 Resultados experimento 5

Instancias clasificadas correctamente	1797
Instancias clasificadas incorrectamente	504
Porcentaje de clasificación	78.0965%
Número de nodos	35
Número de hojas (nodos terminales)	39

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.11 Matriz de confusión del experimento 5

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1444	35
Satisfecho	469	353

La representación gráfica del árbol es muy por lo que se omite.

4.1.6 EXPERIMENTO 6

Los resultados del experimento número 6 son los siguientes:

Tabla 4.12 Resultados experimento 6

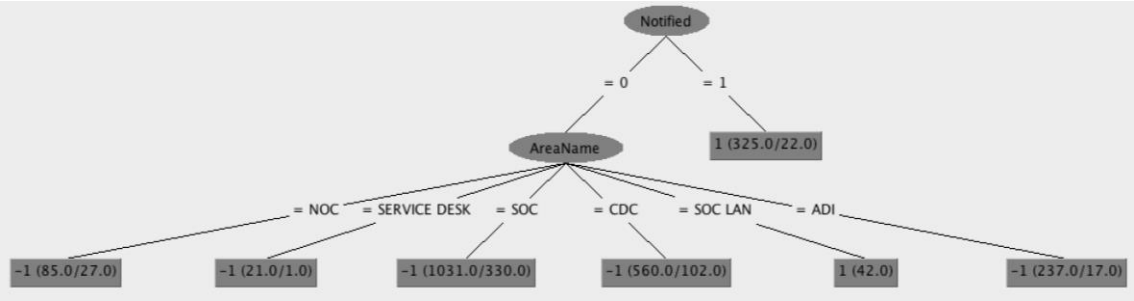
Instancias clasificadas correctamente	1803
Instancias clasificadas incorrectamente	498
Porcentaje de clasificación	78.3572%
Número de nodos	7
Número de hojas (nodos terminales)	9

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.13 Matriz de confusión del experimento 6

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1448	31
Satisfecho	467	355

Figura 4.4 Árbol de decisión generado en el experimento 6



4.1.7 EXPERIMENTO 7

Los resultados del experimento número 7 son los siguientes:

Tabla 4.14 Resultados experimento 7

Instancias clasificadas correctamente	1803
Instancias clasificadas incorrectamente	498
Porcentaje de clasificación	78.3572%
Número de nodos	35
Número de hojas (nodos terminales)	39

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.15 Matriz de confusión del experimento 7

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1433	46
Satisfecho	452	370

La representación gráfica del árbol es muy grande por lo que se omite.

4.1.8 EXPERIMENTO 8

Los resultados del experimento número 8 son los siguientes:

Tabla 4.16 Resultados experimento 8

Instancias clasificadas correctamente	1800
Instancias clasificadas incorrectamente	501
Porcentaje de clasificación	78.2269%
Número de nodos	35
Número de hojas (nodos terminales)	39

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.17 Matriz de confusión del experimento 8

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1425	54
Satisfecho	447	375

La representación gráfica del árbol es muy grande por lo que se omite.

4.1.9 EXPERIMENTO 9

Los resultados del experimento número 9 son los siguientes:

Tabla 4.18 Resultados experimento 9

Instancias clasificadas correctamente	1805
Instancias clasificadas incorrectamente	496
Porcentaje de clasificación	78.4442%
Número de nodos	42
Número de hojas (nodos terminales)	50

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.19 Matriz de confusión del experimento 9

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1430	49
Satisfecho	447	375

La representación gráfica del árbol es muy grande por lo que se omite.

4.1.10 EXPERIMENTO 10

Los resultados del experimento número 10 son los siguientes:

Tabla 4.20 Resultados experimento 10

Instancias clasificadas correctamente	1792
Instancias clasificadas incorrectamente	509
Porcentaje de clasificación	77.8792%
Número de nodos	64
Número de hojas (nodos terminales)	70

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.21 Matriz de confusión del experimento 10

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1388	91
Satisfecho	418	404

La representación gráfica del árbol es muy grande por lo que se omite.

4.1.11 EXPERIMENTO 11

Los resultados del experimento número 11 son los siguientes:

Tabla 4.22 Resultados experimento 11

Instancias clasificadas correctamente	1795
Instancias clasificadas incorrectamente	506
Porcentaje de clasificación	78.0096%
Número de nodos	87
Número de hojas (nodos terminales)	98

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.23 Matriz de confusión del experimento 11

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1387	92
Satisfecho	414	408

La representación gráfica del árbol es muy grande por lo que se omite.

4.1.12 EXPERIMENTO 12

Los resultados del experimento número 12 son los siguientes:

Tabla 4.24 Resultados experimento 12

Instancias clasificadas correctamente	1793
Instancias clasificadas incorrectamente	508
Porcentaje de clasificación	77.9226%
Número de nodos	63
Número de hojas (nodos terminales)	73

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.25 Matriz de confusión del experimento 12

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1399	80
Satisfecho	428	394

La representación gráfica del árbol es muy grande por lo que se omite.

4.1.13 EXPERIMENTO 13

Los resultados del experimento número 13 son los siguientes:

Tabla 4.26 Resultados experimento 13

Instancias clasificadas correctamente	1772
Instancias clasificadas incorrectamente	529
Porcentaje de clasificación	77.01%
Número de nodos	317
Número de hojas (nodos terminales)	343

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.27 Matriz de confusión del experimento 13

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1316	163
Satisfecho	366	456

La representación gráfica del árbol es muy grande por lo que se omite.

4.1.14 EXPERIMENTO 14

Los resultados del experimento número 14 son los siguientes:

Tabla 4.28 Resultados experimento 14

Instancias clasificadas correctamente	1796
Instancias clasificadas incorrectamente	505
Porcentaje de clasificación	78.053%
Número de nodos	108
Número de hojas (nodos terminales)	120

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.29 Matriz de confusión del experimento 14

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1374	105
Satisfecho	400	422

La representación gráfica del árbol es muy grande por lo que se omite.

4.1.15 EXPERIMENTO 15

Los resultados del experimento número 15 son los siguientes:

Tabla 4.30 Resultados experimento 15

Instancias clasificadas correctamente	1798
Instancias clasificadas incorrectamente	503
Porcentaje de clasificación	78.1399%
Número de nodos	108
Número de hojas (nodos terminales)	121

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.31 Matriz de confusión del experimento 15

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1372	107
Satisfecho	396	426

La representación gráfica del árbol es muy grande por lo que se omite.

4.1.16 EJECUCIÓN DEL ALGORITMO ZERO R

El algoritmo Zero – R predice la clase de salida como la clase mayoritaría en el conjunto de datos. El resultado de la ejecución de este algoritmo da los siguientes resultados.

Tabla 4.32 Resultados ejecución Zero R

Instancias clasificadas correctamente	1479
Instancias clasificadas incorrectamente	822
Porcentaje de clasificación	64.2764 %

La matriz de confusión para el modelo generado es la siguiente:

Tabla 4.33 Matriz de confusión del algoritmo Zero R

Clase real / Clase predicha	Insatisfecho	Satisfecho
Insatisfecho	1479	0
Satisfecho	822	0

4.2 ANÁLISIS DE RESULTADOS.

A continuación se condensan los resultados obtenidos en los experimentos.

Tabla 4.34 Condensado de resultados

Número de experimento	Factor de confianza	Número mínimo de objetos por	Precisión	Número mínimo de hojas del árbol

		hoja		
1	0.1	2	77.3577	9
2	0.1	5	77.3577	9
3	0.1	10	77.3577	9
4	0.2	2	78.053	39
5	0.2	5	78.0965	39
6	0.2	10	78.3572	9
7	0.3	2	78.3572	39
8	0.3	5	78.2269	39
9	0.3	10	78.4442	50
10	0.4	2	77.8792	70
11	0.4	5	78.0096	98
12	0.4	10	77.9226	73
13	0.5	2	77.01	343
14	0.5	5	78.053	120
15	0.5	10	78.1399	121
Zero R	-	-	64.2764	

Las conclusiones primarias que se obtiene a partir del condensado son:

- El mejor porcentaje de clasificación se consigue con los parámetros factor de confianza igual a 0.3 y número mínimo de hojas por árbol de 10.

- La precisón lograda con cualquiera de los árboles de decisión generados sobrepasa ampliamente a la obtenida precisión obtenida con el algoritmo de clase dominante.

- La variación de parámetros lograr mejoras en la precisón del árbol de decisión. Sin embargo, éstas no son muy significativas.

Para observar el comportamiento que las modificaciones en los parámetros tienen en la precisión y en el tamaño (complejidad) del árbol de decisión generado, se realizaron las siguientes gráficas.

Figura 4.5 Relación entre el factor de confianza y la precisión del clasificador

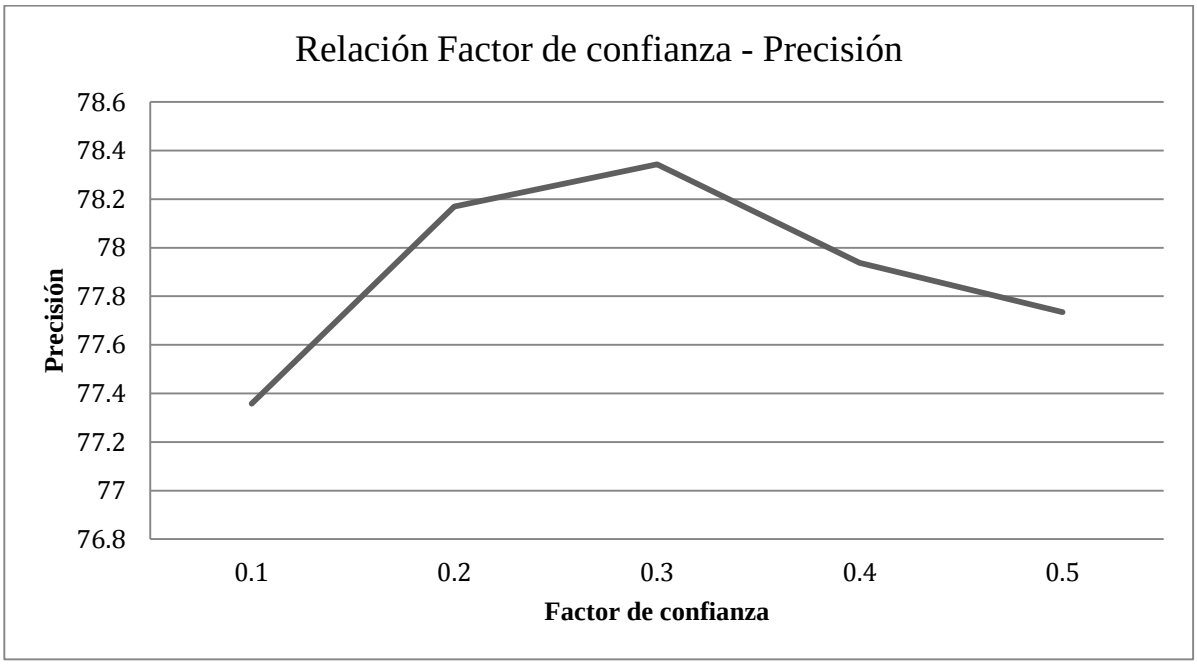


Figura 4.6 Relación entre el factor de confianza y del tamaño del árbol

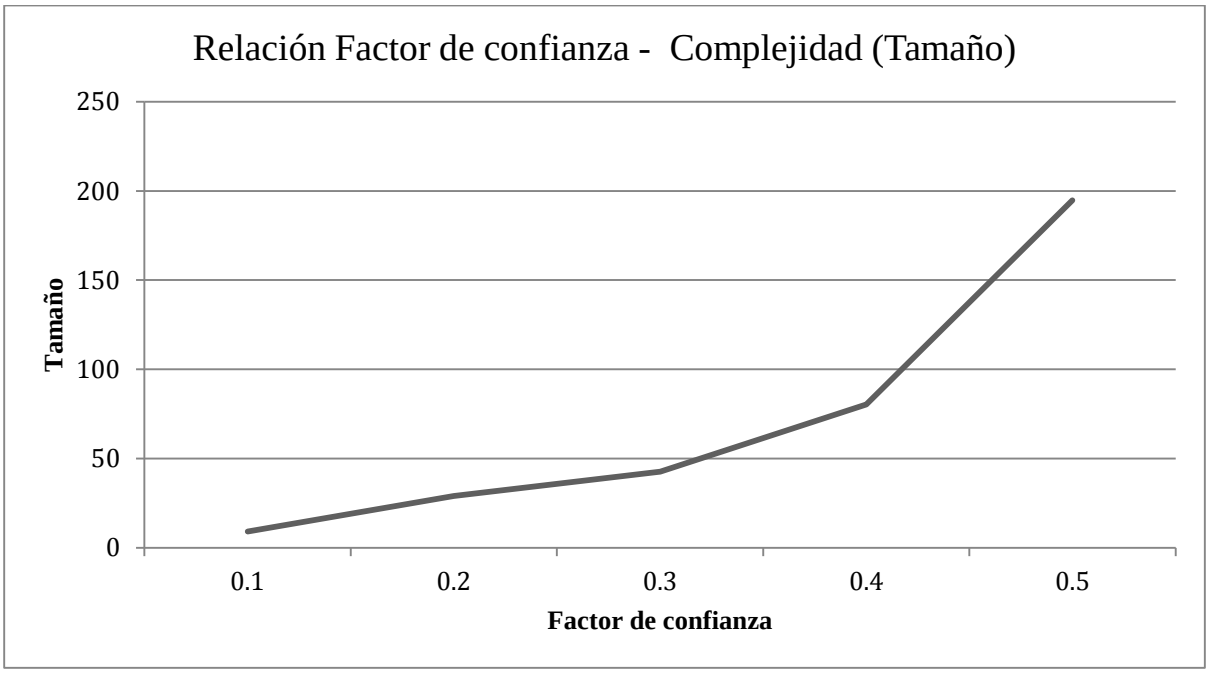


Figura 4.7 Relación entre el número mínimo de objetos por hoja y la precisión del clasificador

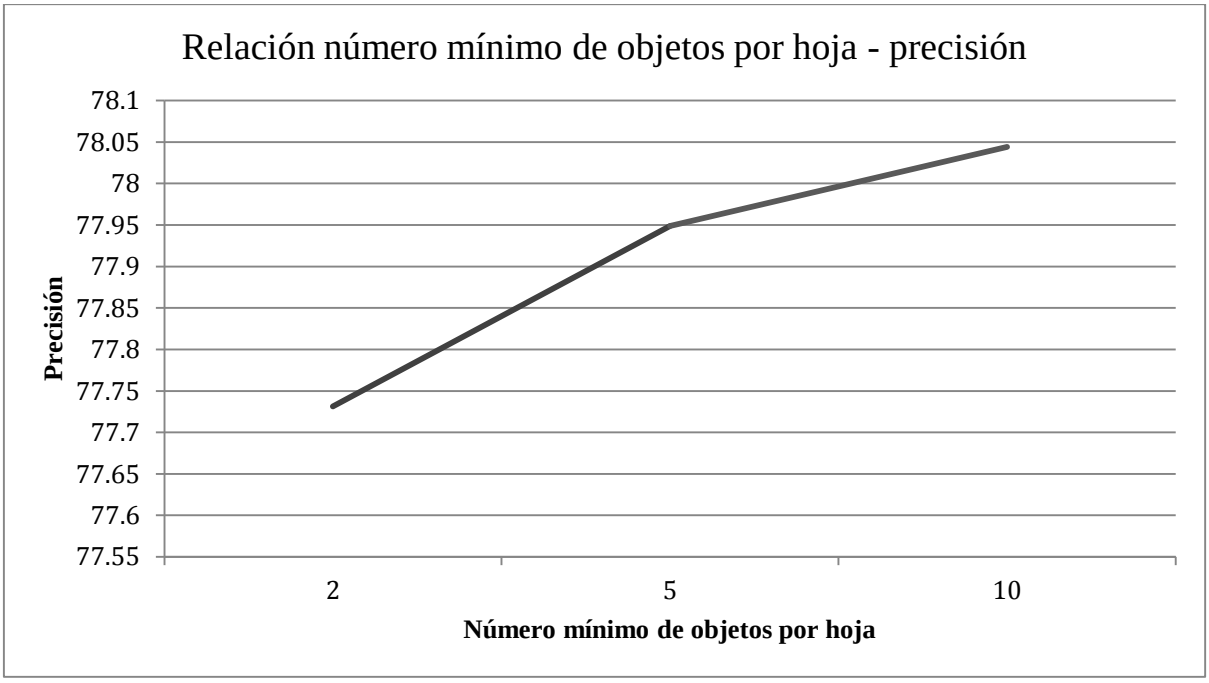
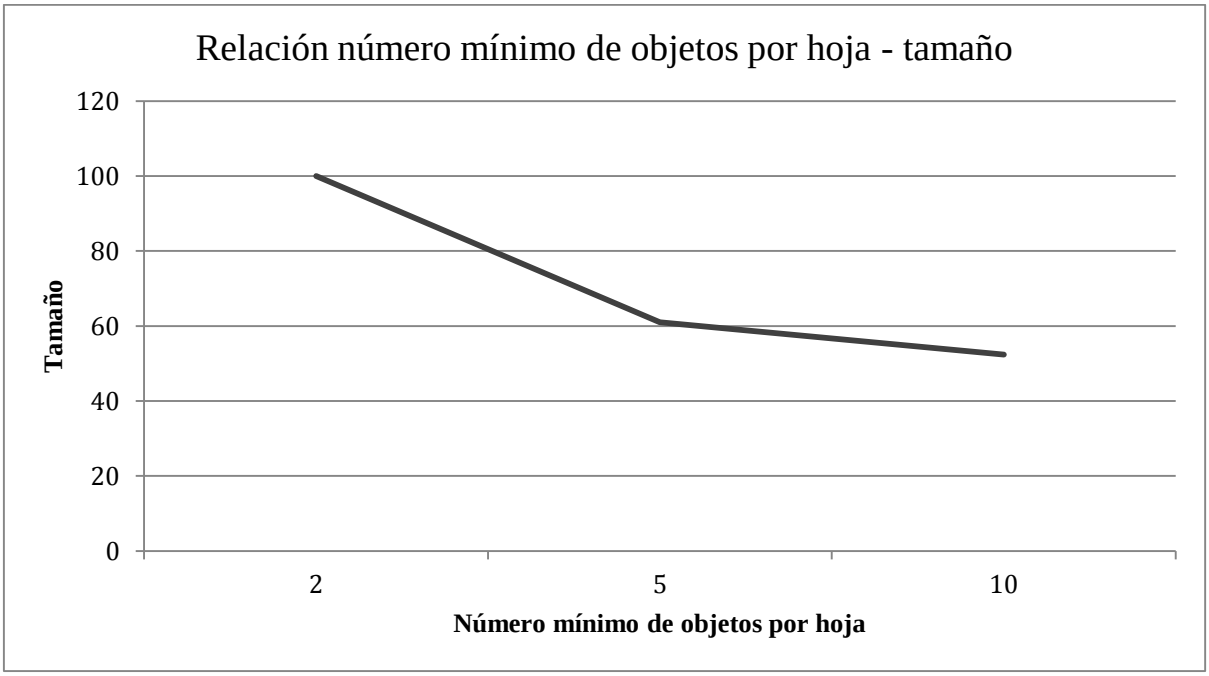


Figura 4.8 Relación entre el número mínimo de objetos por hoja y el tamaño del árbol



A partir de la observación de las gráficas, es posible darnos cuenta de que las variaciones en los valores de ambos parámetros de ejecución tienen efectos importantes tanto en la precisión del árbol de decisión como en su tamaño, y de igual forma, apreciar que como se menciona en la literatura, el tamaño del árbol, directamente proporcional a su complejidad, es inversamente proporcional a su precisión.

Como se mencionó anteriormente, el factor de confianza tiene un impacto directo en al cantidad de poda posterior a la construcción del árbol de decisión, a menor índice de confianza mayor es la poda que se realiza en el árbol, lo que disminiye su tamaño y puede mejorar su precisión. En los experimentos realizados se obtuvo que el mejor valor del factor de confianza es de 0.30. Sin embargo, es importante hacer notar que la diferencia en la precisión entre el mejor y peor valor obtenido es marginal, siendo de 1%.

En la gráfica se observa que después del mejor valor de la precisión, alcanzada con un factor de confianza de 0.3, la precisión comienza de decaer, esto se debe a que se realiza menos poda en el árbol y por lo tanto su capacidad de generalización disminuye.

Por otra parte, el número mínimo de nodos por hoja afecta el grado de poda que se realiza durante la construcción del árbol, previniendo la generación de nuevas hojas y ramas si no cumplen con una cantidad base de elementos; por lo tanto, mientras mayor sea el número mínimo de objetos por hojas el árbol tiene un menor tamaño y su precisión puede aumentar. Sin embargo, es importante que este número no sea demasiado alto porque podría prevenir la generación de ramas u hojas importantes y ser contraproducente para la precisión del árbol.

A partir de cada árbol de decisión generado se pueden determinar una serie de reglas en un formato más entendible para el personal de Mexis y que aportan datos relevantes sobre el proceso de registro.

Uno de los descubrimientos más relevantes que se obtienen al observar el modelo es que el envío de notificaciones al cliente es uno de los atributos más importantes en la percepción del servicio por parte del cliente y por lo tanto en su grado de satisfacción con la transacción.

La siguiente tabla enumera algunas de las reglas que se obtuvieron a partir del mejor modelo encontrado y que fueron validadas por expertos de Mexis:

Tabla 4.33 Reglas de clasificación obtenidas.

Regla	Satisfacción predicha para la transacción
No se notificó al cliente de la existencia de un caso y el área que atiende es el Centro de operaciones de redes	Insatisfecho
No se notificó al cliente de la existencia de un caso y el área que atiende es la Mesa de Servicio	Insatisfecho
No se notificó al cliente de la existencia de un caso, el área que atiende es el Centro de Datos,	Satisfecho

el cliente se encuentra en la Ciudad de México y el área que levantó el caso es Centro de datos	
No se notificó al cliente de la existencia de un caso, el área que atiende es el Centro de datos, el cliente se encuentra en un área distinta a la Ciudad de Mexico (Monterrey, Guadalajara, Puebla)	Insatisfecho
No se notificó al cliente de la existencia de un caso y el área que atiende es el Centro de operaciones de Seguridad de Redes de Área Local	Satisfecho
No se notificó al cliente de la existencia de un caso y el área que atiende es el Administración de incidentes	Insatisfecho
Se notificó al cliente de la ejecución de un servicio.	Satisfecho

5 CONCLUSIONES Y TRABAJO FUTURO

5.1 CONCLUSIONES

El trabajo desarrollado cumplió el objetivo básico de generar un árbol de decisión con una índice de clasificación mayor al obtenido con el algoritmo Zero R, ya que se obtuvo un modelo con una precisión de 78.4442%, que representa una mejora de 22% con respecto a la precisión de 64.2764% alcanzada con el algoritmo Zero R.

Adicionalmente, se obtuvieron reglas a partir del mejor modelo encontrado que pueden ser usadas para mejorar el proceso de atención al cliente, como la valoración de las notificaciones al cliente como el atributo más determinante en la satisfacción del cliente al final de la transacción, así como patrones geográficos. Es importante notar que el tipo de solicitud de servicio del caso no fue marcado como relevante por el árbol de decisión para determinar la satisfacción del ticket, sin embargo, la categoría de la solicitud, que involucra a varios tipos si es determinante en algunas reglas. También se observó que existen algunos patrones temporales que pueden incidir en el grado de satisfacción de la transcacción. El área que atiende el ticket es también factor importante en la predicción del modelo.

Sin embargo, aunque se construyó un modelo con una precisión que cumple el objetivo planteado, es importante que se realicen entrenamientos posteriores con una mayor cantidad de datos, lo que puede conducir a la creación de un modelo con aún mejores características.

5.2 TRABAJO FUTURO

A continuación se muestran algunas de las directrices que pueden usar para desarrollar futuros trabajos sobre la generación de modelos predictivos.

5.2 AUTOAPRENDIZAJE

El clasificador construido utilizó como conjunto de entrenamiento los registros del sistema de seguimiento de casos que ocurrieron entre el día 1 de noviembre de 2011 al 18 de julio de 2012. Estos datos representan sólo un subconjunto de los datos totales que existen en el sistema ya que este es alimentado continuamente con nuevos registros, por lo que la actualización del modelo usando los datos no se contemplaron en un entrenamiento anterior, en forma incremental, puede mejorar la precisión del clasificador puesto que se contaría con una mayor cantidad de datos. Sin embargo, es importante hacer notar que se espera que el comportamiento de los usuarios y del personal de atención cambie con el uso del clasificador, lo que implicaría que los patrones contenidos en la información actual cambien. Por lo anterior, es necesario encontrar una combinación de pesos en el conjunto de entrada que le de mayor importancia a los nuevos datos

en el sistema, sin ignorar los datos antiguos. Determinar el peso que debe tener cada instancia del conjunto de entrenamiento es un problema que debe ser considerado para su estudio.

Así mismo, dado que el clasificador ha sido integrado en una aplicación empresarial, es importante que el proceso de autoaprendizaje sea realizado sin afectar la disponibilidad, estabilidad y veracidad del sistema.

5.2 USO DE ALGORITMOS GENÉTICOS

Para este trabajo, la selección de los parámetros de ejecución del algoritmo J48 se realizaron de manera manual, probando diferentes combinación de valores para los parámetros *ConfidenceFactor* y *minNumObj*. Se propone que como trabajo futuro el cálculo de los mejores valores para estos parámetros se realice usando meta – heurísticas como recocido simulado. También es recomendable realizar pruebas con algoritmos genéticos para optimización de valores reales, teniendo como función objetivo la precisión del clasificador. Sin embargo, el incoveniente con el uso de algoritmos genéticos es que se tiene que generar un árbol por cada miembro de la generación, lo cuál eleva el tiempo de cómputo y el uso de memoria, por lo que se deben trabajar en métodos que minimicen el número de árboles generados, usando por ejemplo, memorias temporales. Otro enfoque que usa computación evolutiva es construir el clasificador de manera tradicional, usando el algoritmo J48, y posteriormente evolucionar el árbol de decisión usando programación genética.

5.3 COMBINACIÓN DE LOS DATOS DEL SISTEMA DE SEGUIMIENTO DE CASOS CON OTRAS FUENTES DE INFORMACIÓN

El trabajo realizado utilizó únicamente datos provenientes del sistema de seguimiento de casos. El uso de otras fuentes de información no fue posible ya que no se contaba en ese momento con infraestructura adecuada para obtener la información de estas fuentes bajo demanda, por ejemplo, no era posible obtener indicadores de tráfico actual, tiempo promedio de espera, tiempo promedio antes de colgar o duración promedio de la llamada del sistema de llamadas, ni era posible asociar una llamada con un caso de atención; igualmente, se desconocen algunos atributos de la relación particular de un cliente con Mexis, como acuerdos de nivel de servicio por cliente o información de cobranza.

Sin embargo, se está trabajando en un sistema que integre estas fuentes de información, por lo que posteriormente puede construirse un nuevo clasificador que cuente con más atributos de entrada, lo que haría posible generar modelos más precisos.

REFERENCIAS

[1] GUPTA, Sunil. ZEITHAML, Valarie. "Customer metrics and their impact on financial performance". *Marketing science*. 2006, vol 25, núm. 6, p. 625 - 638

[2] KANNING, Peter. BERGMANN, Nina. "Predictors of customer satisfaction: testing the clasical paradigms". *Managing service quality*. 2009, vol 19, núm. 4, p. 377 - 390.

[3] DAVIS, M. HEINEKE, J. "How disconfirmation, perception and actual waiting times impact customer satisfaction". *International Journal of Service Industry Management.* 1998, vol 9, p. 64 - 73.

[4] ANDERSON, Eric. "Long-run effects of promotion depth on new versus established customers: Three field studies". *Marketing sci.* 2004, vol 23, núm. 1, p. 4 - 20.

[5] RUCCI, Anthony. KIRN, Steven. QUINN, Richard. "The employee - customer – profit chain at Sears". *Harvard bus.* 1998

[6] ITIL. *IT Service Management Zone.* [ref. de 1 de octubre 2012]. Disponible en web: http://www.itil.org.uk/sm-goal.htm

[7] BORBÓN, Jeffrey. *¿SOC? si, Security Operations Center.* [ref. de 15 de octubre de 2012] Disponible en web: http://hacking.mx/seguridad/%C2%BFsoc-si-security-operations-center

[8] BMC Software. *Footprints family.* [ref. de 2 de octubre 2012]. Disponible en web: http://www.numarasoftware.com/footprints/

[10] HANCOCK, T. *et al*. Lower bounds on learning decision list and trees. *Information and Computation.*,1996, Vol. 2, Nº 126, p. 114–122

[11] QUINLAN, J. Machine learning 1, 1986.

[12] QUINLAN, J. C4.5 Programs for Machine Learning, Morgan Kaufmann, 1993.

[13] BREIMAN, L. *et al* .Classification and Regression Trees. Wadsworth Int. Group, 1984.

[14] BERRY, M. LINOFF, G. Mastering data mining: The art and science of customer relationship management. Wiley, 2000.

[15] HILL, N. ALEXANDER, J. The handbook of customer satisfaction and loyalty measurement. Gower House, 2006. 273 p.

[16]TRUXILLO, C. et al. Advanced Business Analytics, Volume 1. SAS Institute, 2010.

[17] DRAZIN, Sam. MONTAG, Matt. "Decision tree analysis using Weka". Universidad de Miami.

[18] JIAWEI, Han. KAMBER, Micheline. Data mining: Concepts and techniques. 2ª edición. Morgan Kaufmann. 2006.

[19] HAND, David. MANNILA, Heikki. SMYTH, Padhraic. Principles of Data mining (Adaptative Computation and Machine Learning). Bradford. 2011

[20] KANTARDZIC, Mehmed. Data Mining: Concepts, Models, Methods, and Algorithms. 1ª edición. Wiley-IEEE Press. 2002.

[21] MAIMON, Oded. ROKACH, Lior. Data Mining and Knowledge Discovery Handbook. 1ª edición. Springer. 2005.

[22] The University of Waikato. Weka 3: Data Mining Software in Java. [ref. de 1 de octubre 2012]. Disponible en web: http://www.cs.waikato.ac.nz/ml/weka/

[23] RUD, Olivia. Data mining cookbook: modeling data for marketing, risk and customer relationship management. Wiley. 2001

[24] BERRY, Michael. LINOFF, Gordon. Data mining techniques: for marketing, sales and customer support. Wileyand sons. 1997.