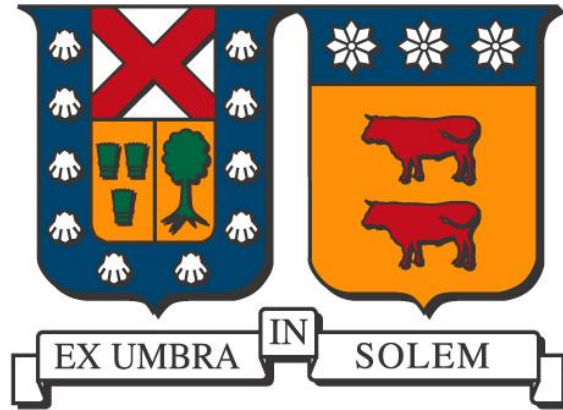


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO – CHILE



**CREACIÓN DE MODELOS PREDICTIVOS PARA
LA DEMANDA DE CLIENTES DE UNA EMPRESA
DEL RUBRO TI**

IGNACIO ANDRÉS RODRÍGUEZ GACITÚA

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERÍA
CIVIL INFORMÁTICA**

PROFESOR GUÍA :

JOSÉ LUIS MARTÍ LARA

PROFESOR CORREFERENTE :

CECILIA REYES COVARRUBIAS

Mayo 2017

Agradecimientos

Quiero dar gracias en primer lugar a mi familia que ha sido un apoyo fundamental en estos años universitarios. A mi padre quién fue un momento de alegría cuando la cosas se dificultaban. A mi madre que me ha inculcado el sentido de responsabilidad y la fortaleza para aspirar a ser más.

Quiero agradecer profundamente a Almendra Salgado quién siempre estuvo en los mejores y peores momentos, agradezco que me obligara a ir a clases. Siempre creíste en mí incluso en los momentos en donde mi confianza se vio más vulnerada. Su cariño y ayuda fueron claves en mi éxito universitario.

A mis hermanos, Camila, José Tomás y Jaime Pablo les agradezco que soportarán mi mal humor durante todos estos años. En especial quiero agradecer a José y Jaime quienes me ayudaron a realizar en parte algunos cálculos de éste trabajo . A mi abuela Silvia del Carmen Gandulfo le agradezco toda su paciencia, atención y cariño, no sólo en mi vida universitaria sino durante toda mi vida. A mis mascotas Otto y Fito por pasar noches en vela trabajando conmigo.

Quiero sentidamente agradecer al Profesor José Luis Martí por todos los buenos consejos que me otorgó en el transcurso de la carrera, por el gran apoyo que me dio en éste último proceso de mi vida universitaria y por sobre todo por la dedicación y rigurosidad con la que trabaja.

Finalmente a todos los amigos que conocí, a los profesores y funcionarios de la universidad gracias por hacer de éste proceso una muy linda parte de mi vida.

Resumen

El presente trabajo se centra en un proyecto de inteligencia de negocios con la finalidad de poder predecir la demanda de los distintos segmentos de clientes que posee Ki Technology, una empresa dedicada a la creación de soluciones digitales.

Para la realización de este proyecto se utilizaron tecnologías e infraestructura de *big data*, específicamente *Apache Spark*, con las cuales se crea una arquitectura capaz de procesar y administrar grandes cantidades de datos, permitiendo la creación de modelos predictivos flexibles e independientes de la cantidad de datos.

Los resultados obtenidos corresponden a cinco modelos distintos, los cuales se construyeron principalmente bajo las técnicas de *Gradient Boosted Tree* y *Decision Tree*, debido a que éstos presentaron los mejores rendimientos, en base a las métricas propias de las técnicas de regresión.

Abstract

This final work is a business intelligence project in order to make a prediction of the Ki Technology client's demands. To do this we used the historical data of the Company related with the behavior of their clients.

Big data tool are used to accomplish the goals of this Project, in particular Apache Spark was used as the main framework. This feature makes this work flexible and capable to process big amount of data without a problem.

The results are 5 different models. Those models were built with this two algorithms Gradient Boosted Tree and Decision Tree. Those models have a high quality predictions, that is very useful and will be very helpful for the decisionmaking.

Índice de Contenidos

Capítulo 1: Descripción del problema.....	1
1.1. Conocer al Cliente y Estacionalidad	1
1.2. Detección del Problema	2
1.3. Ki Teknology y Propuesta de Solución.....	3
1.4. Objetivos de la memoria	4
1.5. Resultados esperados de la memoria.....	5
Capítulo 2: Estado del Arte	6
2.1. Metodologías para proyectos de minería de datos.	6
2.1.1. Metodología CRISP-DM.....	6
2.1.2. Metodología <i>KDD</i>	8
2.1.3. Metodología SEMMA.....	9
2.2. Aprendizaje Automático	11
2.3. Algoritmos de minería de datos	12
2.4. <i>Big Data</i>	13
2.4.1. <i>Apache Hadoop</i>	14
2.4.2. <i>Apache Spark</i>	15
2.4.3. Apache Spark vs Apache Hadoop.....	17
Capítulo 3: Propuesta de Solución	19
3.1. Comprensión del negocio.....	19
3.1.1. Determinación de los objetivos del negocio.....	19
3.1.2. Valoración de la solución.....	21
3.1.3. Determinación de los objetivos de minería de datos.....	21
3.1.4. Realización de un plan de proyecto.....	22
3.2. Comprensión de los datos	24
3.2.1. Recolección de los datos iniciales	24
3.2.2. Descripción de los datos.....	26

3.2.3.	Exploración de los datos	27
3.2.4.	Verificación de la calidad de los datos.....	29
3.3.	Preparación de los datos.....	30
3.3.1.	Selección de los datos	30
3.3.2.	Limpieza de los datos.....	30
3.3.3.	Estructuración de los datos.....	31
3.3.4.	Integración los datos.....	31
3.3.5.	Formateo de los datos.....	32
Capítulo 4:	Desarrollo y validación de la solución propuesta.....	34
4.1.	Modelado.....	34
4.1.1.	Selección de la técnica de modelado	34
4.1.2.	Generación de un plan de prueba	35
4.1.3.	Construcción del modelo.....	37
4.2.	Evaluación de los resultados	41
4.3.	Proceso de revisión.....	46
4.4.	Determinación de futuras fases	48
4.5.	Plan de Implementación	48
4.6.	Plan de monitoreo y mantención.....	50
4.7.	Revisión del proyecto.....	50
Conclusiones		53
Bibliografía		56
Anexos.....		58
Modelo Aerolíneas		58

Índice de Figuras

Figura 1: Ciclo de la metodología CRISP-DM	8
Figura 2: Etapas que componen el proceso KDD	9
Figura 3: Etapas de la metodología SEMMA	10
Figura 4: Ejemplo de regresión lineal, para representar la dependencia entre ventas y gasto en publicidad	13
Figura 5: Arquitectura de almacenamiento HDFS	15
Figura 6: Núcleo de Apache Spark	16
Figura 7: Esquema de distribución de trabajos en Apache Spark	17
Figura 8: Ejemplo de conexión a la base de datos	25
Figura 9: Ejemplo de consulta para la replicación de los datos	25
Figura 10: Demanda general de todos los clientes de Ki Tecnologia	28
Figura 11: Demanda particular de un cliente de Ki Tecnologia	29
Figura 12: Data mart generado para la integración de datos	33
Figura 13: Formateo de los datos, para la conversión del vector de características.....	33
Figura 14: Reportes realizados en QlikView	41

Índice de Tablas

Tabla 1: Comparación entre Hadoop y Spark	18
Tabla 2: Plan de proyecto para la implementación	23
Tabla 3: Descripción de los datos de la tabla ISSUES	26
Tabla 4: Descripción de los datos tabla PROYECTOS	27
Tabla 5: Descripción de los datos de la tabla TIMES_ENTRIES	27
Tabla 6: Resultado de la verificación de datos	29
Tabla 7: Datos seleccionados para la creación del modelo	30
Tabla 8: Clase que alberga el data set de entrenamiento	32
Tabla 9: Clase que alberga el data set de prueba	32
Tabla 10: Tabla generada por Spark con los resultados del modelo	33
Tabla 11: Clasificación del coeficiente de correlación	36
Tabla 12: Graduación de los parámetros de Gradient Boosted Tree	38
Tabla 13: Graduación de los parámetros de Decision Tree Regressor	38
Tabla 14: Graduación de parámetros de Random Forest	38
Tabla 15: Graduación de parámetros de LinearRegression	39
Tabla 16: Comparación en los resultados de algunas pruebas con Gradient Boosted Tree	39
Tabla 17: Comparación en los resultados de Decision Tree Regressor	39
Tabla 18: Comparación en los resultados con Random Forest	39
Tabla 19: Selección de los parámetros que presentan los mejores resultados para cada algoritmo	40
Tabla 20: Porcentaje de importancia de cada variable	40
Tabla 21: Resumen de métricas Instituciones financieras	42
Tabla 22: Muestra de resultados de Instituciones financieras con Decision Tree Regressor	42
Tabla 23: Resumen de métricas Aerolíneas	43
Tabla 24: Muestra de resultados de Aerolíneas con Decision Tree Regressor	43
Tabla 25: Resumen de métricas Mercado automotriz	43
Tabla 26: Muestra de resultados del Mercado automotriz con Decision Tree Regressor	44
Tabla 27: Resumen de métricas Empresas alta demanda	45
Tabla 28: Muestra de resultados de Empresas de alta demanda con Gradient Boosted Tree	45
Tabla 29: Resumen de métricas Empresas de baja demanda	45
Tabla 30: Muestra de resultados de Empresas de baja demanda con Decision Tree Regressor	46

Introducción

Este trabajo forma parte de una investigación realizada para la empresa Ki Teknology, la cual tiene como objetivo predecir la demanda de los clientes de esta empresa orientada a la elaboración de soluciones tecnológicas, tales como desarrollo de software a medida, desarrollo de aplicaciones web y mantenimiento de sistemas, entre otras. De esta forma Ki Teknology podrá generar nuevas oportunidades de negocio, además de conocer de mejor manera el comportamiento de sus clientes a lo largo del tiempo. Para llevar a cabo esto se utilizaron los datos de gestión de la empresa, los que actualmente no están siendo explotados como fuente de información para futuros negocios.

Para la realización de este trabajo se implementaron técnicas de *machine learning*, usando la metodología CRISP-DM, la cual fue la guía para la elaboración de este proyecto. Por otro lado, como se quiere fomentar la utilización de los modelos predictivos en la empresa, se construyó una arquitectura de *big data* para el almacenamiento y procesamiento de los datos. Con esto se garantiza la flexibilidad y escalabilidad de este proyecto, con la intención de que perdure en el tiempo.

Para que el conocimiento encontrado por el modelo sea útil, debe ser presentado en forma de reportes, lo cual ayuda a la comprensión y legibilidad. De esta forma los resultados son de gran utilidad para el apoyo en la toma de decisiones.

El informe a desarrollar se estructura en cuatro capítulos. El capítulo 1 corresponde a la descripción del problema; se contextualiza el ámbito de Ki Teknology y cuáles son los problemas que presentan en estos momentos. Con esto se describe el problema en profundidad. Se fijan los objetivos tanto del negocio como técnicos que esta memoria tiene, como también sus alcances. Finalmente se detalla lo que se espera obtener al finalizar este trabajo.

En el capítulo 2 se realiza un estado del arte. Contiene un marco teórico donde se definen algunos conceptos que serán tratados durante el desarrollo de la memoria. Además, se evalúa el desarrollo de los componentes con los que se trabajarán y como éstos han ido evolucionando a través del tiempo, como también se explican por qué son los más adecuados para el cumplimiento de los objetivos propuestos.

El capítulo 3 se centra en el diseño de la solución, siguiendo la estructura de la metodología CRISP-DM, creada para la realización de un proyecto de minería de dato. Es así que se habla de la descripción de los datos a utilizar, la integración de éstos en la arquitectura de *big data* y del desarrollo de los modelos predictivos. Se finaliza con una evaluación del proceso de comienzo a fin, sin analizar la calidad de los resultados, la que se deja para el capítulo 4.

En el capítulo 4 se lleva a cabo la evaluación de los resultados. Para esto se usan métricas con la intención de validar la calidad de las predicciones obtenidas de los distintos modelos. Se realiza también un proceso de revisión del proyecto en su totalidad para verificar el cumplimiento de los objetivos. Finalmente se determinan algunas mejoras o fases que se puedan implementar en el futuro.

Por último, se entregan las conclusiones obtenidas tras el trabajo realizado, además de las futuras mejoras que se pueden implementar.

Capítulo 1: Descripción del problema

Actualmente, existen pocas empresas que no tengan algún sistema informático que controle o regule algún proceso del negocio; debido a esto es necesario contar con profesionales capaces de desarrollar y mantener los sistemas informáticos en el transcurso del tiempo. Es así que muchas de las empresas decidieron crear nuevas áreas dentro de la organización, las cuales se encargarían de lidiar con todo lo relacionado a las tecnologías de la información. No obstante, no todas las empresas tienen los recursos necesarios para crear dicha área o simplemente, por temas de costos, es más conveniente externalizar esta labor.

La externalización de los servicios informáticos es bastante frecuente y viene influenciada con el desarrollo exponencial que ha experimentado la ciencia de la computación en las últimas décadas. Las empresas cada vez confían más en los resultados de sus proyectos elaborados en sus sistemas de información. De esta manera surge una nueva necesidad en el mercado y con su nacimiento florece una oportunidad de negocio para varias empresas. El desarrollo de software a medida, la administración de la información en sistemas de bases de datos, el marketing comercial a través de internet e incluso el servicio de *hosting* son algunas de las más típicas demandas que conlleva esta necesidad informática. Es así como nace la variada gama de empresas que llegan a suplir esta carencia. La metodología de trabajo es similar; cada labor que es requerida deriva en un proyecto a realizar por parte de la empresa externa, la cual debe comprometerse en los procesos de negocios de la empresa solicitante. Como el compromiso se basa en la elaboración de proyectos por parte de la empresa externa, la gestión de éstos es crucial en el éxito de esta nueva oferta en el mercado.

1.1. Conocer al Cliente y Estacionalidad

Uno de los principales componentes del éxito en una empresa de *outsourcing* es la capacidad de entregar lo que el cliente les solicita en los tiempos establecidos; pero esto no es un gran elemento diferenciador dentro de este mercado. El comprender y conocer al cliente sí lo es. Generalmente los clientes pueden ser clasificados según un sinnúmero de parámetros, ya sea por segmentación en base a sus características, como también por su relevancia para el negocio. Lo importante es poder conocer al cliente de tal manera que la empresa sea capaz de adelantarse a sus necesidades, es ahí donde nace la pregunta ¿Cómo podemos conocer a nuestros clientes?

Existen variadas técnicas que logran este cometido, algunas son más orientadas a lo técnico del cliente, mientras que otras apuntan al lado emocional de éste. Las técnicas orientadas a comprender la parte emocional se centran en conectar con el cliente, siendo la más conocida el mapa de la empatía, el cual se enfoca en segmentar, humanizar y empatizar con el cliente. Para los propósitos de este trabajo se concentrará en la comprensión del cliente mediante los datos duros que se poseen de él. Para esto se ocupará una parte del análisis RFM (Recencia, Frecuencia, Monto) [1], el cual consiste en la medición de las compras recientes de los clientes, la frecuencia de las mismas compras y el monto de éstas. Para cada uno de estos factores se le asigna una puntuación de 1 a 5, siendo los mejores clientes aquéllos que obtengan un 5 de calificación para cada ítem, y los peores quienes tengan un 1 en cada ítem.

El problema en cuestión es que la puntuación puede ponderar no de la mejor manera cada factor, de hecho, la frecuencia de éste puede ser no muy revelador. El monto de las compras y las recientes compras son bastante fáciles de medir, pero la frecuencia tiene un elemento complejo como es el tiempo. Siempre existirán *peaks* de ventas para cada cliente, pero esto no dirá mucho del comportamiento de éste, ya que es sabido que, por ejemplo, en temporadas como navidad las ventas aumentan en casi cualquier segmento del comercio. Pero lo que no se sabe es la estacionalidad de los clientes durante el año. Cada empresa tiene temporadas en el año que son cruciales y específicas, un ejemplo claro es que una línea aérea tendrá una demanda más alta tanto en el verano como en las vacaciones de invierno, por lo que requerirán servicios que apoyen sus propios *peaks* de demanda.

1.2. Detección del Problema

Actualmente no existen muchos elementos diferenciadores que permitan tomar una ventaja en el mercado de empresas TI. Es por esto que la relación con el cliente es fundamental para que la empresa prospere en el futuro y mantenga una rica cartera que logre perdurar en el tiempo. Pero para que la cartera de clientes logre prevalecer es crucial que la empresa conozca al cliente de tal manera que sepa lo que éste le va a solicitar y sobre todo cuándo se lo va a solicitar. Con esta ventaja de tiempo se pueden ofrecer ofertas o mejoras de servicios, anticipándose a la competencia y presentando una clara preocupación por las necesidades de los clientes.

Se necesita conocer la estacionalidad de la cartera de los clientes, es decir, se requiere saber cuándo hay una alta probabilidad de que un cliente requiera un servicio. Lamentablemente la mayoría de los sistemas informáticos no están confeccionados para descubrir el patrón de comportamiento de los clientes, más bien están orientados al buen funcionamiento de los procesos

internos de la empresa, ya sea registrar los datos de los clientes así como la información transaccional de sus ventas, particularmente a fuentes de datos actuales de Ki Teknology no están confeccionadas para poder realizar un análisis que otorgue conocimiento valioso sobre posibles oportunidades de negocios.

De esta forma los problemas detectados son:

1. Nulo conocimiento de la estacionalidad de los clientes.
2. Sistemas de información confeccionados exclusivamente para almacenar datos y no para analizarlos.

1.3. Ki Teknology y Propuesta de Solución

Ki Teknology es una empresa dedicada al rubro de las tecnologías de la información, la cual se dedica a desarrollar soluciones digitales para una amplia gama de clientes. Con más de 20 años de experiencia y más de 1500 proyectos realizados, Ki Teknology representa una de las empresas de *outsourcing* con mayor prestigio dentro del mercado nacional. Dentro de los servicios que ofrece destacan el desarrollo de aplicaciones *web* y móviles, desarrollo de software, consultorías y servicios de *hosting*.

Con la experiencia y los datos de los múltiples proyectos realizados por Ki Teknology se cuenta con una sólida fuente de datos que permite el estudio de los proyectos TI realizados en Chile, lo que será el puntapié para la creación de un sistema que apoye la toma de decisiones en la elaboración y planificación de nuevos proyectos en el futuro. La gestión correcta de un proyecto puede ser un elemento diferenciador con respecto a la competencia, y Ki Teknology quiere ocupar este recurso con el fin de consolidar aún más su nombre en territorio nacional. Es por esto que ve la oportunidad de utilizar las herramientas de *machine learning* y *big data* como un medio para lograr dicha consolidación. Así se espera que la evolución del negocio sea aún más próspera en el futuro al adelantarse tanto a sus competidores como a las necesidades de sus clientes.

Actualmente Ki Teknology posee un sistema de base de datos que registra los datos de la gestión de sus proyectos elaborados en forma de *tickets*. Estos registros almacenan la información más relevante sobre la ejecución de un proyecto, como por ejemplo el tipo de proyecto, las personas asignadas a realizar las actividades, el porcentaje de cumplimiento y las horas trabajadas entre otros aspectos. Sin embargo, esta información no está siendo utilizada para realizar un análisis predictivo que pueda ser de utilidad para la empresa. Este análisis le permitiría conocer a Ki Teknology la naturaleza del proyecto, lo que conlleva conocimiento relevante como la ocurrencia

con que es solicitado cierto tipo de proyecto, cuántas horas hombres son necesarias para cada tipo de proyectos o cómo se distribuye el trabajo generalmente en el tiempo.

La generación de estos *tickets* es de suma importancia ya que, si bien permite llevar un control en la gestión de sus proyectos, agrega un componente que logra la elaboración de posibles ofertas a sus clientes. El tiempo en que estos *tickets* son generados está en relación a la demanda del cliente, de esta forma se puede saber en qué parte del año es cuando se tiene una demanda para determinado cliente. Con esto se puede ir esbozando la estacionalidad de la cartera de clientes de Ki Teknology.

En base a los datos históricos que poseen, la solución a proponer consiste en la creación de un modelo predictivo que utilice datos con el fin de predecir la demanda de su cartera de clientes; con esto se ayuda a gestionar mejor la planificación de un posible proyecto, en base al estudio de estacionalidad. Así Ki Teknology podrá generar nuevas oportunidades de negocio, con la intención de adelantarse a las necesidades, creando una ventaja competitiva en el mercado.

1.4. Objetivos de la memoria

Se desarrollará un nuevo sistema, con el cual la empresa pueda apoyarse el proceso de toma de decisiones, con el fin de elaborar planificaciones de futuros proyectos basadas en los ya realizados. Este modelo predictivo se construirá en un ambiente de *big data*; así, que independiente de los registros con que se cuenten ahora o con los que contarán en el futuro, el modelo presentará consistencia, escalabilidad y rapidez sin importar el tamaño de datos. Este ambiente de *big data* permite también la escalabilidad de este trabajo para la creación de futuros trabajos que quiera realizar la empresa bajo este ambiente.

Objetivo general:

Creación de un modelo predictivo que prediga la demanda de cada cliente en el futuro con el fin de adelantarse a las necesidades de éstos y de esta manera generar nuevas oportunidades comerciales.

Objetivos específicos:

1. Determinar una metodología de proyectos de minería de datos para guiar el desarrollo del trabajo, logrando así estructurar el trabajo bajo un estándar internacional.

2. Generar un ambiente de *big data* para almacenar y procesar de los datos, para que Ki Teknology cuente con un ambiente de última tecnología que se adapte a los desafíos de generar conocimiento.
3. Identificar los algoritmos de minería de datos a utilizar para crear el modelo predictivo, para otorgar algoritmos de calidad a la empresa.
4. Identificar metricas para validar la solución otorgada por el modelo, para garantizar una solución de calidad para la empresa.
5. Elaborar un sistema de reportes para entregar resultados obtenidos de forma legible, para mejorar la toma de decisiones con información antes inexistente.

1.5. Resultados esperados de la memoria

Se espera la creación de un modelo predictivo que sea capaz de apoyar en la toma de decisiones, de forma que lo que las predicciones sean consistentes, precisas y coherentes. Por otro lado, se espera la creación un sistema de reportes que muestre los resultados encontrados por el modelo predictivo y que cumpla la característica de ser legible por los usuarios de Ki Teknology. Además, se espera que la robustez del modelo no se vea afectada por la cantidad de datos que necesite procesar, ya que se elaborará en un ambiente de *big data* que permite una gran escalabilidad.

Capítulo 2: Estado del Arte

En este capítulo se describen las distintas técnicas existentes para desarrollar un proyecto de minería de datos. Se repasarán las técnicas CRISP-DM, *Knowledge Discovery in Databases* y SEMMA, las cuales son las más utilizadas para la elaboración de este tipo de trabajo. Con esto se puede identificar las distintas características de cada una, lo que ayudará a elegir la que se adapte de mejor manera a las necesidades del proyecto. Además se describen los algoritmos de regresión, con los cuales se trabajarán a lo largo de todo éste trabajo. Por último se dan a conocer los procesadores de *big data* que se usarán para cumplir con los objetivos.

2.1. Metodologías para proyectos de minería de datos.

La sistematización de los procesos de minería de datos es vital para el desarrollo de un proyecto de esta índole. Cada metodología implementa una planificación y ejecución distinta, es por esto que dependiendo de las características y requerimientos del trabajo es necesario definir correctamente la que se utilizará. En las siguientes subsecciones se detallan tres de las más conocidas técnicas.

2.1.1. Metodología CRISP-DM

CRISP-DM (*Cross Industry Standard Process for Data Mining*) es una guía que consta de seis etapas que garantizan el desarrollo de un proyecto de minería de datos. La flexibilidad del método de CRISP-DM logra sobresalir del resto de sus pares ya que, si bien posee una estructura bien definida, sus fases no necesitan ser ejecutadas obligatoriamente de forma sucesiva. A continuación, se presentan las fases que componen CRISP-DM y una breve descripción de cada una de las actividades a desarrollar en ellas:

1. Comprensión del Negocio: corresponde a la comprensión de los objetivos y la transformación de los requisitos empresariales a requisitos técnicos de minería de datos. Esta fase se compone de 4 actividades:
 - Determinación los objetivos del negocio
 - Valoración de la solución
 - Determinación de los objetivos de minería de datos
 - Realización de un plan de proyecto

2. **Comprensión de los Datos:** es la primera fase que interactúa con los datos, con el fin de familiarizarse con el problema. Se identifica la calidad de los datos y el cómo se relacionan entre sí. Se generan las primeras hipótesis de trabajo. Al igual que la fase anterior, ésta se compone de actividades, las cuales son:
 - Recolección de los datos iniciales
 - Descripción de los datos
 - Exploración de los datos
 - Verificación de la calidad de los datos
3. **Preparación de los Datos:** consiste en la preparación de los datos para poder ser procesados por los algoritmos de minería de datos. Se realizan actividades de limpieza, estructuración y almacenamiento, las que se mencionan a continuación:
 - Selección de los datos
 - Limpieza de los datos
 - Estructuración de los datos
 - Integración de los datos
 - Formateo de los datos
4. **Modelado:** se seleccionan los modelos que mejor se ajusten a los datos y las características del problema; para lograr esto se efectúan las siguientes actividades:
 - Selección de la técnica del modelado
 - Generación de un plan de pruebas
 - Construcción del modelo
5. **Evaluación:** se evalúa el rendimiento del modelo, con los criterios de éxito que se hayan fijado previamente. Es importante mencionar que esto sólo se realiza con los datos con los que se ha entrenado el modelo, por lo que es preciso obtener un valor con los datos finales. Las actividades a seguir para efectuar la fase de evaluación son:
 - Evaluación de los resultados
 - Revisión del proceso
 - Determinación de los próximos pasos

6. Implementación: Cuenta con los artefactos necesarios para la instalación y utilización del modelo, además cuenta con las tareas de monitoreo y mantención necesario para el correcto funcionamiento del modelo. Éstos artefactos son:

- Plan de implementación
- Plan de monitoreo y mantención
- Revisión del proyecto

La figura 1 representa el ciclo de la metodología CRISP-DM.

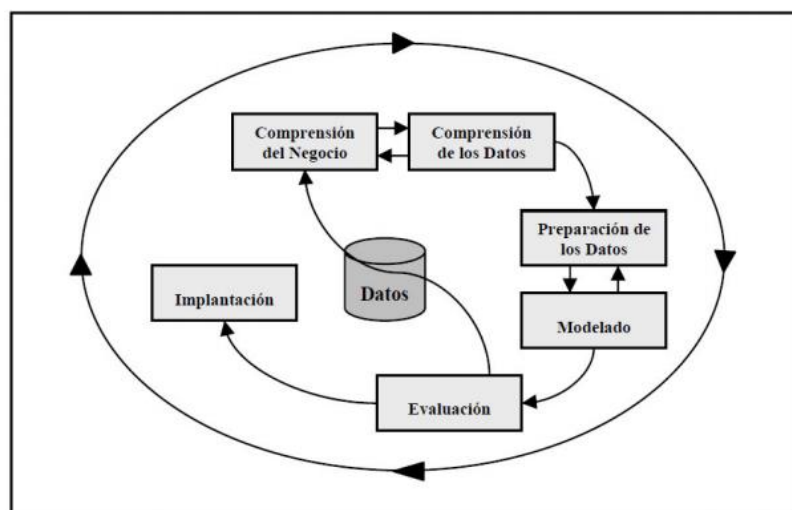


Figura 1: Ciclo de la metodología CRISP-DM [2]

2.1.2. Metodología KDD

Otra de las metodologías más utilizadas en los proyectos de minería de datos es KDD (*Knowledge Discovery in Databases*) la que, al igual que CRISP-DM, tiene como finalidad descubrir conocimiento útil dentro de los repositorios de datos de una empresa. Se caracteriza por ser un proceso iterativo que explora grandes cantidades de datos para determinar relaciones entre éstos. En la figura 2 se pueden apreciar las distintas etapas que contempla el desarrollo de un proyecto bajo esta metodología.

Si bien, existen varias semejanzas con CRISP-DM se refleja la clara ausencia de la comprensión del negocio por parte de KDD, por lo que es considerado un proceso donde la comprensión de los datos ha sido realizada *a priori*. Las etapas de este proceso tienen las siguientes finalidades:

1. Selección de los datos: se determinan los datos a utilizar, como también las fuentes de información de donde se realiza la extracción de estos.
2. Preprocesamiento: es donde se realizan todos los procesos de limpieza de los datos ya extraídos. Generalmente es aquí donde algunos de los datos sufren algunas transformaciones para que el tratamiento posterior sea más fácil de realizar. Habitualmente esta etapa es conocida como procesos ETL.
3. Transformación: todos los cambios que no se realizaron en el paso anterior son llevados a cabo en esta etapa, donde el dato se prepara finalmente para poder ser procesado por algún algoritmo de minería de datos. Las operaciones más comunes realizadas son de agregación de datos o normalización.
4. Minería de datos: corresponde al modelo propiamente tal, en donde los algoritmos son aplicados con el objetivo de encontrar las relaciones que puedan existir.
5. Interpretación y evaluación: finalmente, se identifican los patrones encontrados en la etapa anterior que puedan ser de utilidad. Todo esto bajo métricas de evaluación que deben ser consistente al objetivo de búsqueda.



Figura 2: Etapas que componen el proceso KDD [3]

2.1.3. Metodología SEMMA

Otra de las metodologías más utilizadas es SEMMA (*Sample, Explore, Modify, Model, Assess*), la cual posee la característica de ser bastante más acotada que CRISP-DM, porque al igual que KDD se centra más en el proceso de desarrollo de minería de datos y no en los objetivos empresariales. Sus siglas corresponden a las cinco etapas de la que se compone esta metodología, las que se pueden apreciar en la figura 3.

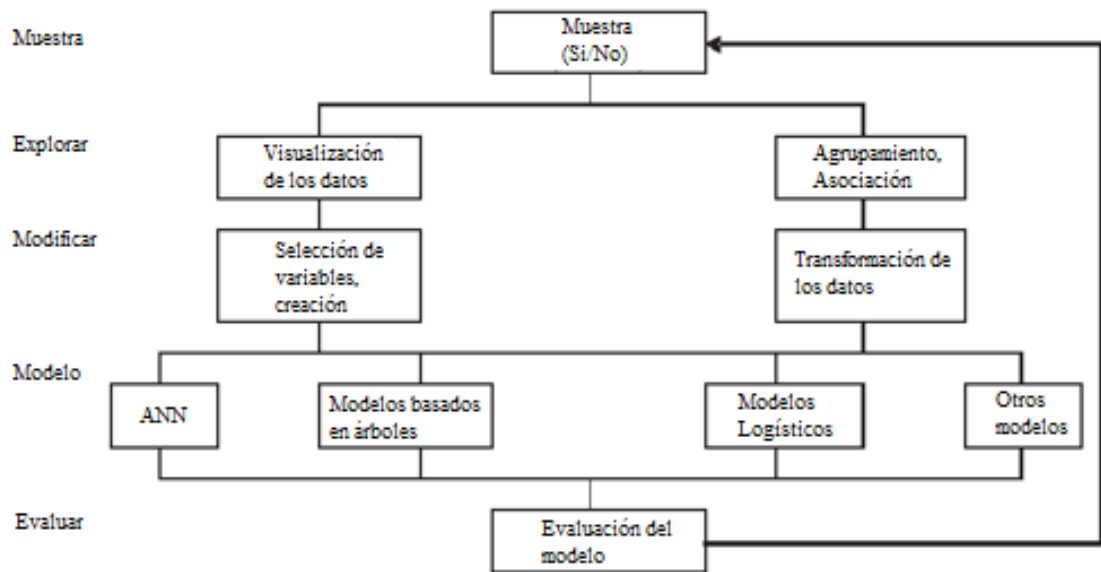


Figura 3: Etapas de la metodología SEMMA [4]

La descripción de cada etapa es la siguiente:

1. **Muestra:** se extrae una proporción de datos que sea una buena muestra de toda la población de datos existentes; esto se realiza con el objetivo de facilitar el proceso de búsqueda sobre los datos, reduciendo el tiempo.
2. **Explorar:** en esta etapa se detectan anomalías en los datos para posteriormente eliminarlos de la muestra extraída. Con esto se ayuda a refinar el proceso de descubrimiento de la información en las siguientes etapas. Se suele utilizar el apoyo de *softwares* que permiten la visualización de los datos.
3. **Modificar:** se aplica la transformación de los datos, que al igual que en la metodología KDD suele ser una modificación o una agregación de información. Esto puede ocurrir en variadas ocasiones, ya que a medida que se obtengan resultados se puede necesitar modificaciones que no fueron incluidas; de esta forma presenta la misma naturaleza iterativa que las anteriores metodologías.
4. **Modelo:** es en esta etapa donde ocurre el descubrimiento de conocimiento y de correlación entre los datos. Aquí es donde la utilización de los algoritmos de minería de datos entran en funcionamiento para la solución del problema.

5. Evaluar: luego de la obtención de los resultados se realiza una evaluación. Se analiza el grado de éxito de lo obtenido de acuerdo a los parámetros de entrada que se fijaron en las primeras entradas. Para la evaluación de los resultados se suele ejecutar todo el proceso de nuevo para muestras distintas, si éste muestra los mismos grados de satisfacción que el primero se valida el modelo.

2.2. Aprendizaje Automático

El aprendizaje automático es una rama de estudio de la Inteligencia artificial que tiene como objetivo lograr que las computadoras aprendan. De esta manera se pueden crear aplicaciones capaces de generalizar el comportamiento a partir de información estructurada. Actualmente, se encuentra inmersa en variadas aplicaciones de la vida cotidiana como avances médicos, motores de búsquedas o detección de fraude con tarjetas de crédito. El resultado del aprendizaje automático comúnmente son modelos matemáticos que describen el comportamiento de los datos o predictivos.

Modelos

Como se mencionó, el aprendizaje automático tiene como objetivo generar modelos como resultado de una tarea dada. En general, es posible distinguir tres tipos de modelos, los cuales se explican a continuación:

1. Modelos geométricos: modelos que pueden tener mas de una dimensión.
2. Modelos probabilísticos: determinan la distribución de probabilidades descriptora de una función que entrelaza valores característicos de los datos.
3. Modelos lógicos: expresan probabilidades de un evento en forma de árboles de decisión.

El aprendizaje automático posee un conjunto de algoritmos que se agrupan según sus funcionalidades y aplicaciones:

1. Aprendizaje supervisado: estos algoritmos producen una función que establece una correspondencia entre las entradas y las salidas. Es comúnmente utilizado para la clasificación de los datos, en donde el algoritmo trata de etiquetar un conjunto de vectores. Se caracteriza por el entrenamiento que se realiza al algoritmo para lograr el aprendizaje deseado. Este entrenamiento debe ser controlado por un agente externo, quien es el encargado de validar la respuesta generada, para los distintos parámetros de entrada.

2. Aprendizaje no supervisado: a diferencia del aprendizaje anterior, se desconocen las clases en las que se debería etiquetar la información; por tanto el sistema debe ser capaz de encontrar por si solo los patrones que logren etiquetar la información. Generalmente es asociado a algoritmos de agrupamiento (*clustering*).
3. Aprendizaje semi-supervisado: es una combinación de los aprendizajes descritos anteriormente; se debe trabajar con datos etiquetados y otros sin etiquetar.
4. Aprendizaje por esfuerzo: el aprendizaje se realiza en base al entorno. Los datos de entrada son utilizados como retroalimentación que se obtiene del mundo exterior, como respuesta a sus acciones. Es un sistema que obtiene el aprendizaje mediante el ensayo-error.

2.3. Algoritmos de minería de datos

Para aprendizaje supervisado existen dos tipos de algoritmos, de clasificación y de regresión. Como se explicará en secciones posteriores, se optará por la utilización de la regresión la cual se adapta más a la solución del problema. A continuación, se presentan algunos de los algoritmos que existen dentro de la gama de la regresión:

1. *Gradient Boosted Tree*: es un algoritmo de regresión que utiliza un ensamble de árboles de decisión de manera secuencial, con el objetivo de minimizar una función de pérdida; se optimizan los resultados de cada árbol, para que el entrenamiento de los árboles siguientes sea realizado de mejor manera. Se ajusta bastante bien para datos que poseen una dispersión cíclica, y que no se adaptan a las técnicas de regresión lineal. Sus resultados suelen parecerse a las regresiones realizadas por las series de tiempo.
2. *DecisionTreeRegressor*: este algoritmo mapea las observaciones de un suceso a conclusiones sobre algún valor sobre el mismo suceso. Se esquematiza con alternativas disponibles, que ayudan a tomar una decisión en base a un conjunto de reglas.
3. *Random Forest*: es la utilización de varios algoritmos para árboles de decisión ejecutados paralelamente, donde cada árbol depende de un vector generado aleatoriamente. La generación de estos árboles no correlacionados se promedia obteniendo un resultado final. Se suele ocupar más para los problemas de clasificación, sin embargo tiene la capacidad de resolver problemas de regresión.



Figura 4: Ejemplo de regresión lineal, para representar la dependencia entre ventas y gasto en publicidad [5]

4. *LinearRegression*: corresponde a un modelo matemático que se usa para encontrar una dependencia entre una variable (predictiva) en base a otras independientes. La predicción se basa en una recta que tiene como objetivo reducir el error cuadrático medio de los residuos del modelo. Funciona bastante bien cuando la dispersión de los datos se centra en base a una recta, como se aprecia en la figura 4.

2.4. Big Data

El término *big data* responde a una cantidad de datos lo suficientemente grande como para poder administrarla y procesarla por los sistemas tradicionales de bases de datos. Es por esto que nacen nuevos sistemas de información y estrategias de administración que permiten integrar nuevos tipos de datos además de los tradicionales. Con las nuevas tecnologías de *big data* se puede procesar y administrar cantidades de información de gran tamaño, de forma eficaz y eficiente [6]. Las características por las que se define el término de *big data* son las siguientes:

1. Volumen: se refiere a la cantidad de datos. Se está frente a la presencia de una gran cantidad de datos, generalmente de una medida que sobrepasa el máximo que soportan los sistemas tradicionales de bases de datos.

2. Velocidad: una de las características más esenciales es la gran velocidad con la que el sistema recibe los datos y la velocidad con la que es capaz de procesarlos.
3. Variedad: tiene la capacidad de contener variados tipos de datos, incluyendo aquéllos que no dispongan de un esquema, es decir datos no estructurados. Por esto el *big data* incluye cualquier tipo de archivos, desde estructurados a imágenes o videos.
4. Valor: los datos tienen un valor intrínseco que aún no ha sido descubierto. Como solución se disponen variadas técnicas cuantitativas para poder encontrar el valor que los datos disponen.
5. Veracidad: el problema con la gran cantidad de datos es saber en qué estado se encuentran los datos que se reciben, es decir la calidad de estos y su disponibilidad en el futuro. Por lo mencionado es que *big data* debe asegurarse de que los datos tengan validez para los objetivos que se pretenden alcanzar.

Para el trabajo con gran volumen de datos existen distintos motores de procesamiento de *big data*, como son los casos de *Hadoop*, el motor de datos más conocido y utilizado en la actualidad, y de *Spark*, usado para la elaboración de este proyecto. Ambos provienen de la fundación de *Apache*, por lo que han sido desarrollados básicamente por los mismos principios; sin embargo, poseen cualidades y características distintas a la hora de realizar el almacenamiento y el procesamiento de los datos.

2.4.1. *Apache Hadoop*

Hadoop es un *framework* que permite el almacenamiento distribuido y el procesamiento de grandes volúmenes de datos [7]. Llega a solucionar la problemática de descubrir conocimiento en ambientes de datos no estructurados y de gran tamaño. El ecosistema que presenta *Hadoop* cuenta con una gama de soluciones, las que apoyan las siguientes funciones:

1. Gestión de datos: dado que el objetivo es almacenar y procesar grandes cantidades de datos, *Hadoop* cuenta con su propio sistema de archivos llamado *Hadoop Distributed File System* (HDFS), el cual permite el almacenamiento de archivos grandes, a través de muchas máquinas distribuidas dentro de un *clúster*. Presenta un mecanismo de fiabilidad de datos mediante la replicación en las diversas máquinas [8].
2. Acceso a datos: una de las ventajas que posee este motor de *big data* es la variedad de formas en las que se puede acceder a los datos almacenados; *Hadoop* provee una gran cantidad de aplicaciones que pueden manipular y acceder a éstos, tales como *Apache Hive* y *Apache Drill* [9].

3. Gobernabilidad e integración de datos: otra de las ventajas es el amplio espectro de aplicaciones que permiten una integración rápida y eficiente de los datos. Las más conocidas son *Apache Falcon*, *Apache Flume* y *Apache Sqoop* [10].

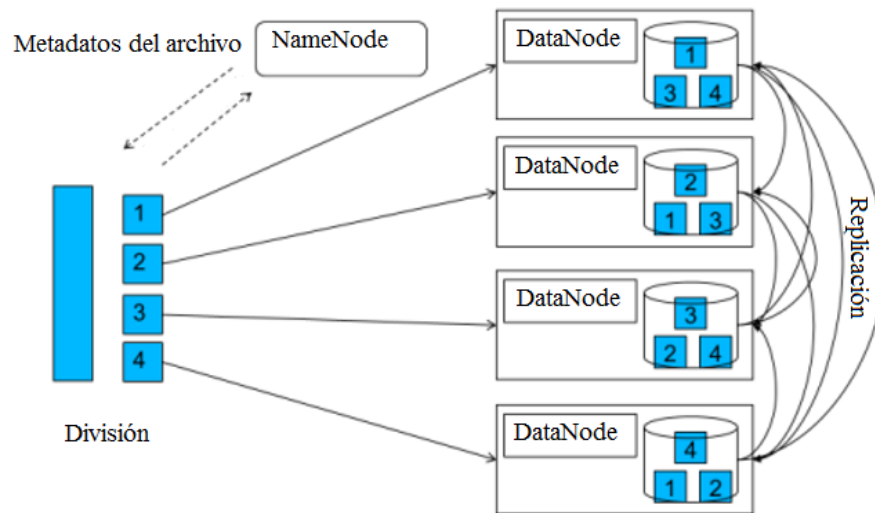


Figura 5: Arquitectura de almacenamiento HDFS [11]

4. Seguridad de la información: se cuenta con *Apache Knox*, una aplicación que se responsabiliza por otorgar un único punto de acceso a los servicios que ofrece *Apache Hadoop*. Así se asegura la simplicidad en el área de la seguridad, tanto para los administradores del *clúster* como también para aquéllos que acceden a los nodos [12].
5. Operaciones: la operación más regular es la ejecución de aplicaciones sobre la arquitectura de *Hadoop*, para esto se cuenta con la ayuda de *Apache ZooKeeper* quien coordina los procesos distribuidos [13].

2.4.2. *Apache Spark*

Apache Spark no sólo es un motor de procesamiento de *big data*, sino que es una completa plataforma elaborada para el tratamiento de esta materia [14]. La gran ventaja que tiene *Spark* sobre *Hadoop* es que fue construido para ejecutar algoritmos iterativos de alta complejidad en la memoria; de esta forma reduce las lecturas y escrituras en disco, generando un procesamiento incluso 100 veces más rápido que el procesamiento de *Hadoop*. Por otra parte *Spark* contiene un

framework integrado que incluye componentes de aprendizaje automático, *streaming*, *SQL* y grafos. Los componentes son:

1. *Spark SQL*: módulo que permite trabajar con datos estructurados. La ventaja es que puede acceder a los datos a través de consultas *SQL*, de esta manera se puede trabajar con motores como *MySQL* o *SQL Server*, y a su vez soporta el trabajo con base de datos no estructuradas, como *Cassandra* o *MongoDB*.
2. *Spark Streaming*: permite la elaboración de aplicaciones altamente tolerables a las fallas, para transmisiones en vivo.
3. *MLlib*: módulo que alberga variadas librerías de *machine learning*.
4. *GraphX*: API que permite el trabajo de grafos complejos y la computación paralela sobre estos.

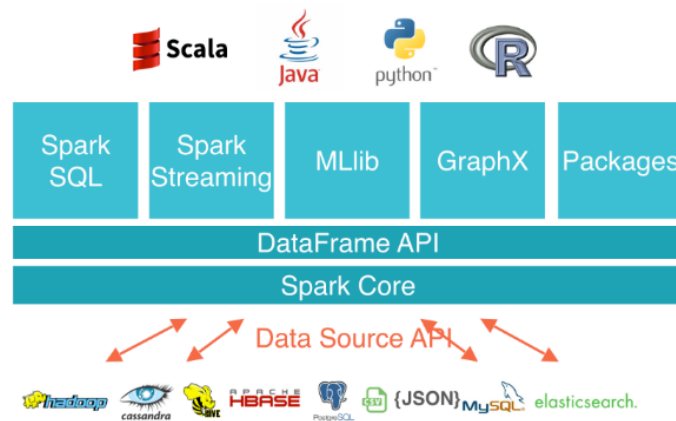


Figura 6: Núcleo de Apache Spark

La estructura de *Spark* se compone de un sistema de computación distribuida en *clústers* de computadores. El funcionamiento de *Spark* es otra de las grandes ventajas que tiene sobre *Hadoop*. Cada tarea *Spark* crea un grafo acíclico dirigido (DAG), es decir, cada etapa de una tarea es representada como componentes (nodos) de un grafo la cual es enviada a un *clúster* para que la procese. Para la elaboración de los distintos nodos o etapas de la tarea, *Spark* cuenta con los *Resilient Distributed Dataset* (RDD) que son la unidad básica del tratamiento de datos, y que permiten realizar las operaciones con grandes cantidades de datos en la memoria del computador. Los RDD sólo aceptan dos tipos de operaciones, las cuales son:

1. Transformaciones: que transforman el RDD actual en un nuevo RDD.
2. Aplicaciones: que se realizan en los RDD con la finalidad de obtener un valor como resultado.

El procesamiento de los datos funciona de forma distribuida en las máquinas que componen el o los *clústers* de ordenadores. Para esto *Spark*, contiene un ordenador que se encarga de distribuir el trabajo de forma que el procesamiento se realice de forma simultánea en cada nodo. El encargado de asignar el trabajo recibe el nombre de Programa controlador y los encargados de ejecutar el trabajo son denominados Nodo trabajador. El funcionamiento es bastante sencillo: el Programa controlador divide el *dataset* en RDD, los cuales son enviados a cada trabajador; además transmite el DAG para que éstos sepan la secuencia de transformaciones que deben realizar a cada RDD. Luego de ejecutar el DAG cada nodo retorna su resultado de vuelta al Programa controlador, el encargado de reunir todas las respuestas, integrarlas y entregar el resultado final.

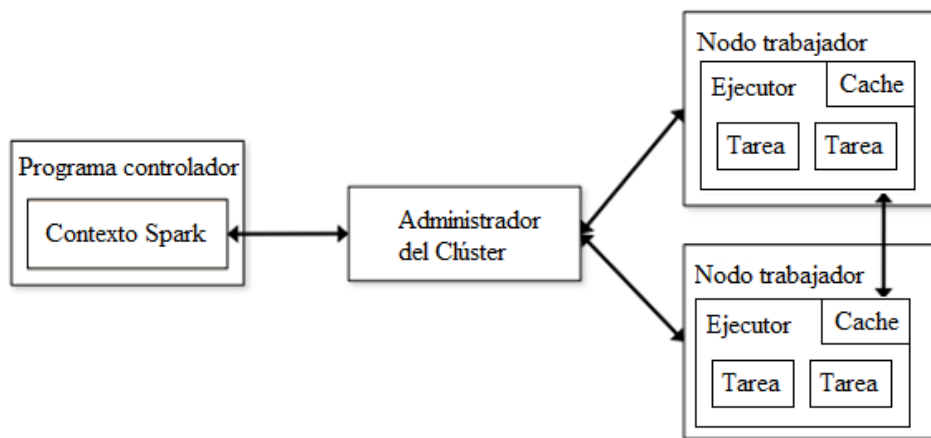


Figura 7: Esquema de distribución de trabajos en Apache Spark

2.4.3. Apache Spark vs Apache Hadoop

Si bien *Spark* no es totalmente independiente de *Hadoop*, sí ha demostrado ser el recambio en el procesamiento de grandes volúmenes de datos. *Spark* no está diseñado para almacenar datos, ya que su principal característica está en el procesamiento de lotes en memoria, con esto se elimina el costo de procesamiento *MapReduce* que identifica a *Hadoop*, lo que genera un gran impacto en la velocidad de procesamiento, logrando que *Spark* procese los datos de manera mucho más veloz. Por otro lado, *Spark* posee su propio administrador de *clúster*, el cual se llama *Standalone*, por lo que no requiere el administrador que por defecto que trae *Hadoop*, el cual es *YARN* [15]. La dependencia que posee con *Hadoop* es la forma en que se realiza el almacenamiento que mantiene el HDFS (*Hadoop File System*). Otra de las características positivas de *Spark* es el núcleo de éste, el cual se compone de variadas bibliotecas que soportan múltiples funciones como minería de datos, *streaming* y consultas *SQL*; esto se diferencia de *Hadoop*, ya que depende de terceros para la utilización de las mismas funcionalidades. De esta manera *Spark* se ha

posicionado como el sucesor de *Hadoop*, presentando significativas mejoras, sin embargo, aún está en constante desarrollo, lo que hace que sea una solución un poco apresurada para aquellas empresas que tienen todo ambientado en *Hadoop*. Sin embargo, para empresas que recién se inician en el mundo del *big data*, resulta ser la mejor herramienta para el desarrollo de sus aplicaciones.

La tabla 1 hace una comparativa de las características entre ambos *frameworks*.

Tabla 1: Comparación entre Hadoop y Spark

	<i>Hadoop</i>	<i>Spark</i>
Arquitectura de almacenamiento	HDFS	HDFS
Administrador de clúster	1. Mesos 2. Yarn	1. <i>Standalone</i> 2. Mesos 3. Yarn
Lugar de almacenamiento	Disco duro	Memoria RAM
Tipo de procesamiento	MapReduce	<i>Directed Acyclic Graph</i>
Seguridad	1. Yarn 2. Knox Gateway 3. Sentry	Yarn
Lenguajes soportados	Java	1. Java 2. Python 3. Scala 4. R
Shell interactiva	No	Sí

Capítulo 3: Propuesta de Solución

Este capítulo contempla la propuesta de solución al problema descrito anteriormente; para esto primero se tiene que comprender la situación actual de la empresa y del negocio mismo. Es por esto que la metodología de minería de datos elegida es CRISP-DM, debido a que se necesita realizar una comprensión del negocio antes de ejecutar la construcción de la solución, que es básicamente lo que la diferencia de las otras dos descritas en el capítulo anterior. Para esto las fases que se desarrollarán de esta metodología son las siguientes:

1. Comprensión del negocio
2. Comprensión de los datos
3. Preparación de los datos
4. Modelado
5. Evaluación

En las siguientes secciones se encuentra el desarrollo de cada una de las fases.

3.1. Comprensión del negocio

La primera fase de la metodología CRISP-DM contempla la comprensión del negocio y la detección de los problemas. Una vez realizado esto se presenta una solución, la cual se desarrollará como una solución de minería de datos. Es probable que ésta sea la fase más importante dentro de la metodología, ya que agrupa tanto precisar objetivos del proyecto como también los requisitos desde la perspectiva empresarial, para luego convertirlos en objetivos técnicos dentro del marco de un plan de trabajo. Sin importar cuales sean las técnicas de minería de datos que se ocupen, si no se conoce el negocio no se obtendrán los resultados deseados. El conocimiento del problema permitirá decidir correctamente cuáles son los datos que permiten el desarrollo correcto del proyecto, como también validar si los resultados obtenidos son exitosos. A continuación, se describe de forma complementaria al capítulo 1 las actividades que se realizan en esta primera fase.

3.1.1. Determinación de los objetivos del negocio

Esta primera tarea tiene como principal objetivo, el definir el problema a solucionar y tener el primer acercamiento de la minería de datos como una posible solución. Es por esto que en el primer capítulo se mencionaron los problemas que surgen cuando no se conoce del todo al cliente, y como los sistemas de información tradicionales no están diseñados para realizar análisis. De

esta forma se busca generar un modelo predictivo que permita conocer las estacionalidades de cada cliente.

Ki Teknology se encuentra en la búsqueda de un modelo predictivo que prediga la demanda de cada miembro en su cartera de clientes. Con los datos históricos que tiene almacenados, posee una fuente abundante en información sin ser explotada. Cuenta con los registros que utilizan para el monitoreo durante el desarrollo de cada actividad; ésta es bastante variada, yendo desde quien tiene asignada una actividad hasta la cantidad de horas que se ha trabajado en ésta. Así se pueden realizar estudios, sobre la naturaleza de cada proyecto como también el desempeño realizado de los equipos de trabajo; pero lo más importante que revela esta información es el tiempo. Cada *ticket* generado está asociado a un proyecto, demandado por un cliente, lo que permite identificar en qué meses del año se han generado más *tickets*, de esta forma se puede visualizar la distribución de demanda de cada cliente durante el año.

Es así como los objetivos del negocio son bastante claros, se quiere ser una de las empresas pioneras en el área de desarrollo de soluciones tecnológicas que pueda predecir la demanda de sus clientes. Para esto se necesita elaborar un modelo predictivo que entregue información de estacionalidad de los clientes, permitiendo la creación de nuevas oportunidades de negocios. Los objetivos técnicos son la creación de un modelo predictivo que permita predecir la demanda de cada cliente de Ki Teknology, establecer un ambiente de *big data* donde sin importar la cantidad de datos que sean procesados el modelo no pierde consistencia, y generar reportes los cuales entreguen la información encontrada por el modelo.

Uno de los aspectos relevantes son los criterios de aceptación del modelo, es decir, cuál es la exactitud de las predicciones realizadas; los resultados deben ser consistentes, coherentes y exactos con lo que se está solicitando. Por consiguiente, un buen criterio de aceptación es que las predicciones para el nuevo proyecto sean parecidas a las características de proyectos pasados, de esta forma un proyecto que es similar a uno en el pasado no debe tener características que difieran mucho; si dichas diferencias son significativas se puede presumir que la predicción no es del todo certera.

Lo anterior no puede sólo quedar en una observación, se necesita un criterio de aceptación cuantitativo que permita evaluar la exactitud medida en valores numéricos, de esta forma se pueden generar umbrales de aceptación de forma más precisa. El modelo se someterá a pruebas con *datasets* de entrenamiento, los cuales ayudarán a que el algoritmo mejore su aprendizaje sobre el negocio, así las predicciones se realizarán con mayor exactitud, logrando alcanzar los umbrales

de aceptación fijados. Cabe destacar que varios algoritmos pueden ser utilizados con el objetivo de encontrar el que obtenga el mejor nivel de exactitud.

3.1.2. Valoración de la solución

Esta tarea consiste en una evaluación que se realiza de la situación antes de que el proceso de minería de datos sea ejecutado. Es transformar de forma práctica todos los objetivos de negocios en objetivos técnicos, para esto es crucial tener una visión panorámica del problema ya que ésta influirá en los beneficios que este trabajo tenga sobre el modelo de negocio.

Actualmente Ki Teknology conserva los datos históricos de la gestión y control de todos los proyectos que han completado, como también los que se encuentran en ejecución. Sin embargo, los datos sólo son usados para gestionar y no se les está dando un uso más práctico, con los cuales puedan aprender a mejorar su gestión o a su vez obtener nuevo conocimiento de sus clientes. Conscientes del poco uso, surge la oportunidad de crear nueva información a partir de los datos históricos almacenados, requiriéndose de herramientas de minería de datos y de procesamiento de datos que en hoy día no disponen.

Es importante recalcar que los beneficios para la empresa son de gran impacto, ya que no sólo mejoran uno de sus principales procesos, sino que también les ayuda a comprender de mejor manera las necesidades de sus clientes, al estudiar el comportamiento de éstos a lo largo del tiempo. Es así como el costo asociado de este trabajo se ve totalmente cubierto por los beneficios comerciales que proveerá. Los costos de la mantención del modelo se ven reflejados si se requiere realizar modificaciones al mismo o al ambiente de *big data*. Por todo lo mencionado se considera que la situación actual es bastante propicia para desarrollar este trabajo y así generar un elemento diferenciador con respecto a la competencia.

3.1.3. Determinación de los objetivos de minería de datos

En esta sección se sientan los objetivos técnicos (u objetivos de minería de datos), los cuales son la transformación de los objetivos del negocio en metas a alcanzar mediante un análisis de minería de datos

Estos objetivos requieren las siguientes actividades:

1. Generar un ambiente de *big data*, que permita el almacenamiento, tratamiento y procesamiento de los datos. Con este ambiente se mantiene la certeza de que no

importando el tamaño de los datos, el modelo mantendrá su flexibilidad debido a su arquitectura.

2. Definir los algoritmos de minería de datos que permitan la creación del modelo predictivo. Como se mencionó en la sección anterior es importante contar con una gama variada de algoritmos que logran encontrar el mejor grado de aceptación para el modelo.
3. Definir las variables que constituirán las bases predictivas para la creación del modelo predictivo.
4. Definir indicadores que validen la solución otorgada por el modelo. Éstos requieren de la participación de los usuarios de Ki Teknology quienes cuentan con la experiencia para validar los resultados del modelo.
5. Elaborar con un *set* de entrenamiento que permita evaluar la solución del modelo.
6. Elaborar un sistema de reportes que permita la lectura de los resultados entregados por el modelo, para posterior difusión a los distintos usuarios.

3.1.4. Realización de un plan de proyecto

Esta tarea tiene como finalidad presentar el plan de proyecto que se seguirá para el desarrollo del modelo predictivo. Se contemplan todas las fases que se requieren para la construcción del modelo predictivo. Este plan de trabajo sólo cubre las actividades técnicas, por lo que apunta más a la manipulación de herramientas de minería de datos. En la tabla 2 se presenta el plan de trabajo asociado.

Este plan de proyecto está construido en base a las tareas necesarias para llevar a cabo al proyecto de forma exitosa. Este plan de trabajo es una herramienta para la empresa con la que se puede apoyar para entender lo que se realizará de una manera resumida y concisa. Detalla los aspectos técnicos de la instalación de la arquitectura, la integración de los datos al ambiente de *big data*, el análisis y elección de los algoritmos a utilizar para la construcción de los modelos predictivos; finalmente contempla la implementación, que incluye también la revisión y la validación de los resultados como también la representación de éstos en el formato de reportería que la empresa requiere.

Tabla 2: Plan de proyecto para la implementación

Implementación del ambiente de Big Data	Se implementa una arquitectura de <i>big data</i>, en particular se ocupará <i>Apache Spark</i>, la cual proveerá la flexibilidad y la consistencia del modelo sin importar el tamaño y cantidad de datos. ○ Instalación de Apache Spark en las maquinas que componen el clúster: se instala <i>Apache Spark</i> en una cierta cantidad de máquinas virtuales, que compondrán el clúster. ○ Instalación de un ambiente de desarrollo Apache Spark: Se instala en el nodo maestro un ambiente de desarrollo (IDE) para la elaboración del modelo. Se utiliza una adaptación de <i>Eclipse</i> para el lenguaje <i>Scala</i>. El proyecto generado incluirá las siguientes dependencias, <i>Apache Spark SQL</i> y <i>Apache Spark MLlib</i>, las cuales son utilizadas para el manejo de los datos que se encuentran en la base de datos de Ki Teknology y para la utilización de herramientas de minería de datos. ○ Instalación de una herramienta de inteligencia de negocios: esta herramienta será la encargada de representar la información encontrada en el modelo en formato de reporte, la herramienta seleccionada es <i>QlikView</i>.
Replicación de datos	Comprende la realización de la replicación de los datos desde la base de datos de Ki Teknology al ambiente de <i>big data</i> generado por <i>Apache Spark</i>. ○ Selección de los datos: se seleccionan los datos que serán utilizados para la elaboración del modelo. ○ Transformación de los datos: los datos seleccionados son transformados con el formato que corresponde a los algoritmos de minería de datos a usar. ○ Creación de un almacén de datos: los datos son replicados en este nuevo almacén de datos para posteriormente poder ser procesados.
Procesamiento y Análisis	Se utilizan los algoritmos de minería de datos que se encuentran en <i>Apache Spark MLlib</i> y la manipulación de los datos a través de <i>Apache Spark SQL</i>, para el procesamiento y creación del modelo predictivo. ○ Selección de las variables críticas para la creación del modelo: se seleccionan las variables que se utilizarán para la predicción de los sucesos.

	○ Elaboración del modelo: con las herramientas de minería de datos se construyen, a través de <i>Apache Spark MLlib</i>, varios modelos predictivos con las variables seleccionadas. ○ Evaluación de los modelos: se evalúan los modelos construidos, con el objetivo de utilizar el que presente el mejor rendimiento.
Implementación	Es la última etapa del proyecto y contempla la implementación del modelo con los datos de prueba de Ki Technology. ○ Evaluación del modelo: con los datos de producción de Ki Technology se evalúan los resultados con un set de pruebas. ○ Representación de los resultados: con la herramienta de inteligencia de negocios, se presentan la información a los usuarios finales.

3.2. Comprensión de los datos

Esta fase corresponde a la comprensión de los datos, la cual comienza con la recolección inicial de los mismos, pasando también por la descripción, la exploración y finalmente por una verificación de la calidad de éstos. Con esto se logra mantener un control total sobre los datos recolectados, lo que es sumamente importante ya que gran parte del éxito en la construcción del modelo se funda en esta fase.

Todos los datos que se utilizarán se encuentran en los repositorios de información en el ambiente de producción de Ki Technology. Se realiza la replicación dentro de la arquitectura de *big data*, para posteriormente describir estos datos. Ejecutando un análisis sobre éstos se puede lograr un mayor entendimiento de la naturaleza de los mismos, para así encontrar una mejor solución al comprender de mejor forma el problema. Finalmente se verifica su calidad, lo que dará paso a la siguiente fase que es la preparación de los datos.

3.2.1. Recolección de los datos iniciales

El principal objetivo de la recolección es obtener todos los datos iniciales que se encuentren en la base de datos de la empresa. Es así como se puede listar aquéllos que serán adquiridos para el futuro procesamiento, se precisarán las tablas de origen de donde se extraerán los datos y los métodos a utilizar para llevar a cabo dicha tarea.

La información se recolectará desde la base de datos de producción de Ki Teknology, principalmente de las siguientes tablas, presente en un modelo multidimensional:

1. ISSUES
2. PROYECTOS
3. TIMES_ENTRIES

siendo ISSUES la tabla principal donde converge la mayoría de datos. En esta tabla se encuentran, principalmente, las actividades que cada integrante de Ki Teknology desarrolla durante un proyecto. Los datos que se encuentra en esta tabla son de naturaleza cualitativa, ya que corresponde a una tabla de hechos. Con esto se monitorea y se gestiona el control de las actividades de un proyecto, debido a que se puede apreciar en tiempo real el desarrollo de una actividad asignada a un miembro de la empresa.

Por otro lado las tablas PROYECTOS y TIMES_ENTRIES son dimensiones por lo que, básicamente, cuentan con información descriptiva de los proyectos y de las actividades, como son la naturaleza del proyecto, los clientes que solicitaron el proyecto y el nombre de las actividades a desarrollar, entre otros descriptores.

Para la recolección de los datos se ocupa *Apache Spark SQL*, que mediante consultas directas sobre la base de datos de Ki Teknology logra obtener lo requerido. Esta conexión se realiza a través de un conector *JDBC*. La consulta ingresada para la conexión a la base de datos es la que se ilustra en la figura 8.

```
val url = "jdbc:mysql://server_name/database_name"
val username = "*****"
val password = "*****"
Class.forName("com.mysql.jdbc.Driver").newInstance()
```

Figura 8: Ejemplo de conexión a la base de datos

Mientras que la replicación de los datos se realiza mediante la consulta representada en la figura 9.

```
val myRDD = new JdbcRDD(sc, () => DriverManager.getConnection(url,username,password),
  "SELECT * FROM ISSUES WHERE ID BETWEEN ? AND ?",1,200,1)
```

Figura 9: Ejemplo de consulta para la replicación de los datos

Los datos son almacenados temporalmente en un objeto de *Apache Spark*, llamado *DataFrame*, el cual simula una tabla relacional, como las convencionales de cualquier motor de base de datos SQL.

3.2.2. Descripción de los datos

Una vez que los datos han sido recolectados, se procede a su descripción. Para realizar esto se identificará el volumen que se ha extraído de la base de datos (número de registros y campos por registros) para proceder a identificar cada campo con una breve descripción.

La tabla más relevante es ISSUES, la cual posee la información respectiva de cada actividad en tiempo real. Dentro de los campos se puede apreciar las personas que han sido asignadas a cada ticket, a qué proyecto está asignado y cuál es el porcentaje de desarrollo de la actividad, entre otros. Guarda la información relevante al control y gestión de las actividades que componen un proyecto. Se presentan los datos de ISSUES con sus respectivos atributos y sus respectivas descripciones en la tabla 3.

Tabla 3: Descripción de los datos de la tabla ISSUES

Atributo	Tipo	Descripción
<i>Id_issues</i>	Numérico	Identificador asociado a cada <i>ticket</i> generado
<i>Tracker_id</i>	Numérico	Identificador del elemento <i>tracker</i>
<i>Status_id</i>	Numérico	Estado en que se encuentra el <i>ticket</i> actualmente
<i>Assigned_to_id</i>	Numérico	Identificador de la persona a quien ha sido encargado el <i>ticket</i>
<i>Priority_id</i>	Numérico	Identificador que representa la prioridad que se le ha asignado al <i>ticket</i>
<i>Author_id</i>	Numérico	Identificador de la persona que ha creado el <i>ticket</i>
<i>Locked_version</i>	Numérico	<i>Label</i> que identifica si el proyecto se encuentra bloqueado
<i>Created_on</i>	Numérico	Fecha en la cual ha sido creado el <i>ticket</i>
<i>Updated_on</i>	Numérico	Fecha en la cual ha sido actualizado el <i>ticket</i>
<i>Start</i>	Numérico	Fecha de cuándo debe iniciar el desarrollo de la actividad
<i>Done_ratio</i>	Numérico	Porcentaje que de desarrollo de la actividad asociada al <i>ticket</i>
<i>Root_id</i>	Numérico	Identificador del padre del <i>issue</i> , si este tiene uno
<i>Lft</i>	Numérico	Ratio del trabajo restante
<i>Parent_id</i>	Numérico	Identificador del ticket padre, si es que existe.

La tabla PROYECTOS contiene la información descriptora de cada proyecto; incluye quienes son los clientes, el tipo de proyecto, las fechas de plazo y presupuesto, entre otros datos. La tabla 4 describe los campos de PROYECTOS.

Tabla 4: Descripción de los datos tabla PROYECTOS

Atributo	Tipo	Descripción
<i>Id</i>	Numérico	Identificador asociado al proyecto
<i>Name</i>	<i>String</i>	Nombre del proyecto
<i>Description</i>	<i>String</i>	Descripción breve sobre la empresa
<i>Homepage</i>	<i>String</i>	Sitio web de la empresa
<i>Is_public</i>	Numérico	Identificador que indicia si el proyecto es de carácter publico
<i>Parent_id</i>	Numérico	Identificador del proyecto padre, si es que existe.
<i>Created_on</i>	Numérico	Fecha en la cual ha sido creado el ticket
<i>Updated_on</i>	Numérico	Fecha en la cual ha sido actualizado el <i>ticket</i>
<i>Indentifier</i>	<i>String</i>	Identificador descriptivo, no numérico
<i>Lft</i>	Numérico	Ratio del trabajo restante

La tabla TIMES_ENTRIES contiene el detalle de los *tickets* generados, con descripciones de las personas encargadas de realizar el trabajo, las horas que se han utilizado en el desarrollo de ese ticket, como también la información relevante a cuando fue creado. En la tabla 5 se muestra la descripción de los datos de TIMES_ENTRIES.

Tabla 5: Descripción de los datos de la tabla TIMES_ENTRIES

Atributo	Tipo	Descripción
<i>Id</i>	Numérico	Identificador asociado al detalle cada <i>ticket</i> generado
<i>Project_id</i>	Numérico	Identificador del proyecto al que está asociado el <i>ticket</i>
<i>User_id</i>	Numérico	Identificador del usuario encargado del <i>ticket</i>
<i>Issue_id</i>	Numérico	Identificador del <i>ticket</i> asociado
<i>Hours</i>	Numérico	Horas trabajadas en el <i>ticket</i>
<i>Parent_id</i>	Numérico	Identificador del ticket padre, si es que existe.
<i>Comments</i>	<i>String</i>	Comentarios sobre la descripción del <i>ticket</i>
<i>Activity_id</i>	Numérico	Actividad realizada
<i>Spent_on</i>	Numérico	Tiempo utilizado
<i>Tyear</i>	Numérico	Año de realización
<i>Tmonth</i>	Numérico	Mes de realización
<i>Tweek</i>	Numérico	Semanada de realización
<i>Created_on</i>	Numérico	Fecha de creación del <i>ticket</i>
<i>Updated_on</i>	Numérico	Fecha de actualización

3.2.3. Exploración de los datos

Esta fase corresponde a una exploración, con el objetivo de describir, visualizar y relacionar los datos; involucra un estudio estadístico de los datos, es decir, se realizan las primeras

visualizaciones en formatos de gráficos, lo cual ayuda a la comprensión del problema y a la elaboración de la solución.

En base a los datos seleccionados se pueden obtener varios gráficos que ayudan a la comprensión del problema. La imagen de la figura 10 es la demanda en general que tiene Ki Teknology en base a todos los proyectos realizados con todas sus empresas, durante un periodo de tiempo que abarca desde 2009 a 2016. A simple vista se puede apreciar el crecimiento que ha experimentado la empresa durante el transcurso del tiempo, siendo un claro ejemplo la curva de demanda del año 2009 en comparación al año 2015. Pero por otro lado también se aprecia que desde el 2013 la demanda ha disminuido paulatinamente, lo que es preocupante por parte del cliente. La tendencia que sale a luz es que el primer semestre siempre ha experimentado una mayor demanda exceptuando el 2012, lo que supone una mayor carga de trabajo para los primeros meses del año.

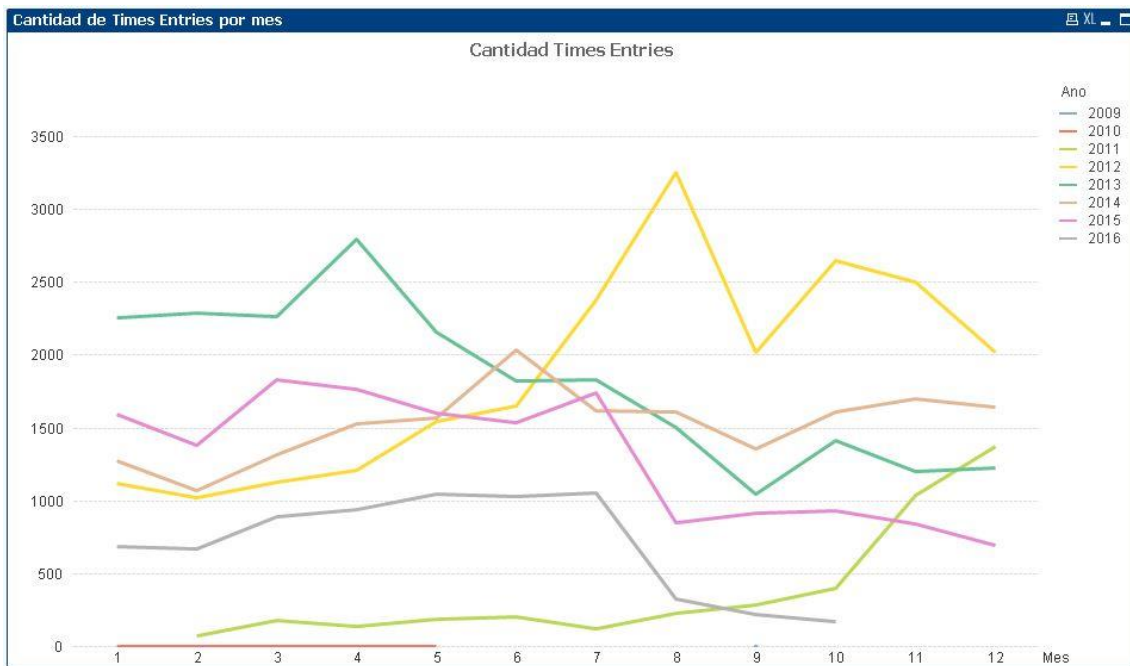


Figura 10: Demanda general de todos los clientes de Ki Teknology

La problemática de la información recabada en la figura 10 es que no se puede distinguir cuales son los clientes que aporten una mayor información, es por esto que se necesita tener la demanda por cada cliente lo que será relevante para determinar la importancia de éste para la compañía. Con una segmentación de los clientes es más fácil desarrollar distintitos modelos predictivos que permitan generar estrategias adecuadas para cada agrupación; dicho de otro modo se puede agrupar la información de todos los clientes en grupos, lo que facilitará tomar decisiones no para cada cliente sino que para cada grupo.

La figura 11 presenta un gráfico de la demanda de uno los clientes de Ki Teknology; esto será sólo para índole informativo de la empresa en caso de querer ejercer una oportunidad para un cliente en particular. Éste es el beneficio que se obtiene con un sistema de reportes que permita una granularidad por empresa, por un lado se puede ofrecer un servicio más especializado y por otro, contaría con la información de forma agrupada para ayudar a identificar qué segmento potenciar y sobre todo cuándo.



Figura 11: Demanda particular de un cliente de Ki Teknology

3.2.4. Verificación de la calidad de los datos

Finalmente está la verificación de la calidad de los datos a utilizar, con esto se quiere verificar la consistencia de los datos. Es así como se logra identificar valores únicos, nulos, fuera de rango etc. De esta forma se reduce el ruido que puede generar estos valores a los algoritmos que se utilizarán. El objetivo principal es asegurar la completitud y corrección de los datos. Los resultados de verificar 86.632 registros en total son los representados en la tabla 6.

Tabla 6: Resultado de la verificación de datos

Verificación de datos	
Valores únicos	12.346
Valores nulos	3.212
Valores fuera de rango	102
Valores mal descritos	7.332

3.3. Preparación de los datos

Esta fase tiene como objetivos seleccionar y preparar los datos para su procesamiento en algún algoritmo de minería de datos. Consiste básicamente en la selección de los datos a utilizar, su limpieza para el procesamiento, la integración con algún almacén de datos y el cambio de formato si es que se requiere para la posterior visualización de los resultados otorgados por el algoritmo.

3.3.1. Selección de los datos

Se selecciona un subconjunto de los datos iniciales recolectados en la fase anterior, apoyándose en los criterios de calidad de los datos que se realizó también en la fase anterior. Estos datos serán los utilizados como entradas para el futuro procesamiento de alguno de los algoritmos de minería de datos escogido para realizar la predicción.

En la tabla 7 se muestran los campos seleccionados, los cuales presumen ser buenos candidatos para el entendimiento del problema.

Tabla 7: Datos seleccionados para la creación del modelo

Datos seleccionados		
Nombre del campo	Tipo de dato	Descripción del dato
Project_id	Int	Identificador que representa a un cliente de Ki Technology
Project.name	Varchar	Nombre de la empresa asociada al identificador anterior
Tyear	Int	Año
Tmonth	Int	Mes
Cantidad_entries	Int	Cantidad de times_entries (<i>tickets</i>)

3.3.2. Limpieza de los datos

Para complementar la fase anterior, se debe optimizar la calidad de los datos con la finalidad de prepararlos para la fase de modelado. Para este caso se requirió exportar los datos en formato de texto para poder hacer una limpieza con un *script* de *Python*.

El gran problema de los datos seleccionados es que sólo se registran aquellos meses en que hubo actividades relacionadas a proyectos. Esto puede influir en las predicciones de los algoritmos

utilizados, ya que al haber meses faltantes, existe la probabilidad de que los resultados se vean afectados por la carencia de más datos.

Los valores que son admitidos por los algoritmos de minería de datos son del tipo *Double* y no del tipo *Int*, por lo que la conversión de los datos numéricos se realizó mediante este *script*. Para mantener la consistencia ningún dato puede tomar un valor nulo, ya que todo el procesamiento debe ser mediante algoritmos que reciban datos numéricos, por lo que parte de la limpieza incluyó la revisión de los datos en busca de valores nulos. En la muestra extraída no se encontraron valores nulos, pero aun así se dejó esta observación para que la implementación a otros *data set* tenga la misma validación y limpieza si es requerida. Finalmente los valores fueron exportados como texto, dicho formato es requerido para ser leído por *Apache Spark*.

3.3.3. Estructuración de los datos

Esta tarea corresponde a la inclusión de operaciones para la preparación de los datos. Para la estructuración de los datos se crearon dos clases en *Apache Spark*; las clases se distinguen entre albergar los *data sets* de entrenamiento y los de prueba. En las tablas 8 y 9 se describen las dos clases en donde se almacenan los datos importados. A diferencia de la clase de entrenamiento, la clase de prueba carece del atributo *label*, ya que es ésta la variable que se va a predecir con el algoritmo de minería de datos.

Las clases mencionadas son la estructura en donde se almacenará la información a nivel de memoria para el procesamiento por parte de los algoritmos de minería de datos, cabe. Para el almacenamiento de los datos de manera física, se construyó un *data mart*, el cual se explica en la siguiente sección.

3.3.4. Integración los datos

La integración de los datos considera la creación de una estructura capaz de almacenar los datos que serán procesados posteriormente. Con esto se procede a la creación del *data mart*, el cual se constituirá de los datos proporcionados en la clase de datos de entrenamiento. Los datos recolectados se componen de las distintas tablas que se seleccionaron en las fases anteriores y de datos calculados. Se diseñará el *data mart* como un modelo estrella, el cual se compondrá de una tabla de hechos y sus respectivas dimensiones. En la figura 12 se pueden apreciar las dimensiones, las cuales son Proyecto, Rubro, Año y Mes mientras que la tabla de hechos se denomina Modelo.

Tabla 8: Clase que alberga el data set de entrenamiento

Clase de datos de entrenamiento		
Nombre de atributo	Tipo de dato	Descripción
Id	<i>Double</i>	Identificador que representa a un cliente de Ki Technology
Nombre	<i>String</i>	Nombre de la empresa asociada al identificador anterior
Año	<i>Double</i>	Año
Mes	<i>Double</i>	Mes
Label	<i>Double</i>	Cantidad de times_entries (<i>tickets</i>)

Tabla 9: Clase que alberga el data set de prueba

Clase de datos de prueba		
Nombre de atributo	Tipo de dato	Descripción
Id	<i>Double</i>	Identificador que representa a un cliente de Ki Technology
Nombre	<i>String</i>	Nombre de la empresa asociada al identificador anterior
Año	<i>Double</i>	Año
Mes	<i>Double</i>	Mes

3.3.5. Formateo de los datos

Consiste en una tarea donde se realizan transformaciones a los datos sin que éstos cambien su significado, con el objetivo de facilitar el procesamiento por alguna técnica de minería de datos. Para este caso el único cambio que se debe realizar es la agrupación de los atributos en forma de vector, es así como los algoritmos son capaces de ejecutar los cálculos necesarios. *Apache Spark* crea una variable del tipo *DataFrame* donde alberga todos los datos como si se tratara de la tabla almacenada en la base de datos. Luego se crea una función que genere el vector de esta forma, el cual se ilustra en la figura 13.

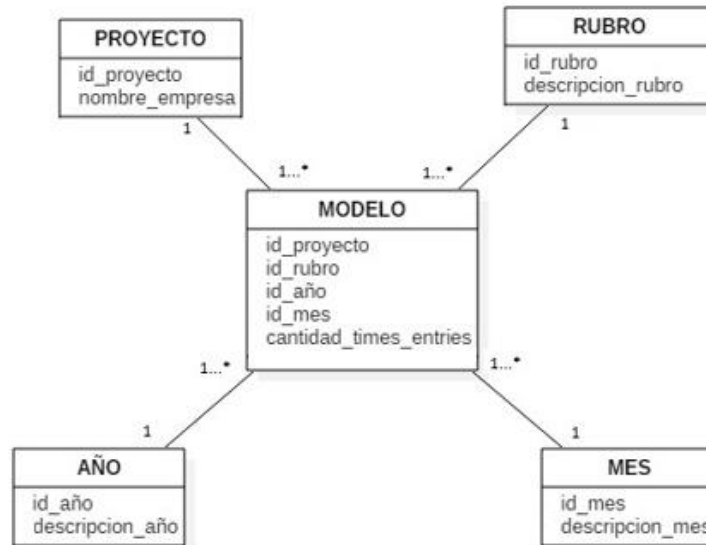


Figura 12: Data mart generado para la integración de datos

```
val assembler = new VectorAssembler()
  .setInputCols(Array("id", "mes"))
  .setOutputCol("features")
```

Figura 13: Formateo de los datos, para la conversión del vector de características

De esta forma las variables que se incluyen en el vector se convierten en las variables que se utilizarán para la generación del modelo predictivo. Con esta transformación de los datos *Apache Spark* genera un nuevo *DataFrame* que contiene la estructura representada en la tabla 10.

Cabe destacar que estos cambios se producen sólo a nivel de memoria, y no son registrados en el *data mart*; el tipo de datos que genera *Apache Spark* es utilizado para el procesamiento de los datos.

Tabla 10: Tabla generada por Spark con los resultados del modelo

Dataframe		
Nombre atributo	Tipo de dato	Descripción
Id	<i>Double</i>	Identificador que representa a un cliente de Ki Technology
Nombre	<i>String</i>	Nombre de la empresa asociada al identificador anterior
Año	<i>Double</i>	Año
Mes	<i>Double</i>	Mes
Label	<i>Double</i>	Cantidad de times_entries (<i>tickets</i>)
Features	<i>Vector</i>	Vector que contiene las variables utilizadas como descriptoras para el modelo predictivo

Capítulo 4: Desarrollo y validación de la solución propuesta

Este capítulo trata de la creación del modelo predictivo. Para llevar a cabo dicha tarea se construye un plan de pruebas que entrega los primeros resultados del modelo. Luego de la construcción y la obtención de los resultados se procede a la evaluación del mismo, con esto se concluyen las fases requeridas por la metodología CRISP-DM para la realización del proyecto. En las siguientes secciones se describen las tareas realizadas para la construcción y posterior evaluación del modelo.

4.1. Modelado

Esta etapa consiste en la elección de las técnicas de modelado más apropiadas para los objetivos de minería de datos. Las técnicas que se utilizarán en esta etapa deben satisfacer las siguientes necesidades:

1. Ser apropiadas al problema
2. Disponer de los datos adecuados
3. Cumplir con los requisitos del problema
4. Contar de un tiempo adecuado para obtener un modelo
5. Conocimiento de la técnica

De la misma manera en que se selecciona la técnica de minería de datos, se debe de escoger un método de evaluación del modelo, que permita obtener un grado de confiabilidad en éste.

4.1.1. Selección de la técnica de modelado

En esta subfase se decide la técnica de minería de datos a utilizar. El problema consiste en la predicción de la demanda de cada cliente de Ki Teknology, a lo largo del año. Lo primero que se debe tener en cuenta es que la variable a predecir representa la cantidad de *tickets*, éstos están asociados a proyectos y en particular consisten en una tarea a realizar. Ésta variable es numérica y continua, es decir no es una predicción del tipo categórica. De esta manera las técnicas que más se ajustan a este tipo de problema son las técnicas de regresión. *Apache Spark* posee variados algoritmos de regresión en su biblioteca de minería de datos, pero se describirán los que tienen mayor soporte dentro de la comunidad de desarrolladores, y los más estudiados y probados para este tipo de problemas.

1. *Gradient Boosted Tree*: es una técnica que sirve tanto para la regresión como para la clasificación. Produce un modelo de predicción en base a variados modelos predictivos

de carácter débil, típicamente en base a árboles de decisión. La construcción de este modelo predictivo se basa en la creación de fases, la que permite una generalización de los datos permitiendo una optimización de los resultados.

2. *Decision Tree Regressor*: también es una técnica en la minería de datos que sirve para la predicción. Si bien está más asociado a la resolución de problemas de clasificación, existen configuraciones de este algoritmo que lo convierte en una fuerte herramienta para predecir variables continuas, logrando que se adapte de gran manera a los problemas de regresión. Se mapea las observaciones de un suceso, con esto se esquematiza las alternativas disponibles, para luego tomar una decisión en base a un conjunto de reglas.
3. *Random Forest*: es una combinación de árboles predictores, que dependen de los valores de un vector aleatorio. Es una variación del *bagging* que se construye en base a una colección de árboles no correlacionados, los cuales se promedian, otorgando el mejor resultado.
4. *LinearRegression*: es un método matemático usado principalmente para identificar la relación entre dos variables, siendo una de éstas dependiente de la otra. Es la técnica más antigua dentro de las que tienen como objetivo minimizar el error cuadrático medio. Esta técnica genera un modelo matemático mediante una fórmula que relaciona las variables independientes, o también conocidas como las variables predictores (en el caso de este trabajo el vector *features*) mediante una combinación lineal con constantes, la cual da como resultado la variable dependiente a predecir.

4.1.2. Generación de un plan de prueba

Ya teniendo el modelo construido se necesita un plan de evaluación. Este plan tiene el objetivo de probar la calidad y la validez del modelo. Pero para poder llevar a cabo esto, primero se debe dividir los datos en dos muestras, una de éstas será utilizada para el entrenamiento de los algoritmos, mientras que la como prueba para corroborar la calidad del modelo.

Las siguientes métricas tienen como principal objetivo validar en qué medida la variable dependiente puede estar explicada por las variables independientes, y estimar la calidad de la ecuación de regresión obtenida. A continuación, se enumeran las métricas que se utilizarán para la medición de la calidad del modelo:

1. Error cuadrático medio (MSE): es un estimador que mide el promedio de los errores de las predicciones al cuadrado. En términos sencillos mide la diferencia entre el estimador y lo que se estima, al cuadrado. Se expresa de la siguiente forma:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2$$

2. Raíz cuadrada del error cuadrático medio (RMSE): mide exactamente lo mismo que la métrica anterior, sólo que no aumenta el error al cuadrado. Se utiliza especialmente cuando no se necesita ser tan preciso con el error; su fórmula es la siguiente:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2}$$

3. Error absoluto medio (MAE): al igual que las métricas anteriores mide qué tan certeras son las predicciones con respecto a los valores reales.

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{f}_i - Y_i|$$

4. Coeficiente de correlación (R^2): es la medida de la relación entre dos variables aleatorias cuantitativas. Es un índice que se utiliza para medir el grado de relación entre las variables estudiadas, se representa de la siguiente forma:

$$R^2 = \frac{\sum x_i y_i - n \bar{x} \bar{y}}{(n-1) s_x s_y}$$

Para el cumplimiento de los objetivos de minería de datos, se debe establecer los valores de aceptación de las métricas de evaluación para verificar la calidad de los resultados, éstas son las siguientes:

- Coeficiente de correlación superior a 70% debido a que se necesita una correlación bien alta y positiva, dentro de las definiciones otorgadas por la metodología CRISP-DM. Las interpretaciones se realizan en base a la tabla 11.
- Menor error cuadrático medio
- Menor raíz del error cuadrático medio
- Menor error absoluto medio

Tabla 11: Clasificación del coeficiente de correlación

Tamaño de la correlación	Interpretación
.90 a 1.0	Correlación bien alta positiva
.70 a .90	Correlación alta positiva
.50 a .70	Correlación moderada positiva
.30 a .50	Correlación baja positiva
0.0 a 0.30	Si existe correlación es pequeña o casi nula

4.1.3. Construcción del modelo

Esta tarea corresponde a la construcción del modelo con las técnicas de minería de datos, descritas en la fase anterior. Se debe señalar que como cada empresa presenta comportamientos distintos, debido a que son empresas de variados rubros, la creación de un único modelo sería una generalización bastante mal lograda. Es por dicha razón que se analizarán las empresas según el rubro e importancia, abarcando la totalidad de las empresas en variados modelos predictivos. Estas son las clasificaciones realizadas:

1. Instituciones Bancarias
2. Aerolíneas
3. Mercado automotriz
4. Empresas de alta demanda
5. Empresas de baja demanda

Cabe destacar que no existe intersección entre estos grupos, es decir ninguna empresa puede estar en dos o más grupos.

Las técnicas que presentaron un mejor alineamiento con la distribución de los datos son *Gradient Boosted Tree*, *Decision Tree Regressor* y *Random Forest*, mientras que *LinearRegression* otorgó los peores resultados para todos los casos.

La selección de los valores para los parámetros de cada algoritmo se debe a las variadas pruebas que se sometieron cada uno. De manera gradual se fue aumentando la combinación de parámetros con la intención de buscar el mejor desempeño. Para *Gradient Boosted Tree* se trabajó con la cantidad de iteraciones, la profundidad de los árboles y la tasa de aprendizaje. Para el primer parámetro se contempló un rango de cinco iteraciones como mínimo y un máximo de sesenta, con una separación de cinco iteraciones en cada prueba. Para la profundidad de los árboles se considera un rango mínimo de tres unidades y un máximo de diez unidades, con una separación de una unidad dentro del rango. Finalmente para la tasa de aprendizaje se estableció un rango de

mínimo de un 20% de aprendizaje y un máximo de un 80%, con una separación equidistante de un 15%. En la tabla número 12 se puede apreciar cuales son los parámetros escogidos para este algoritmo y los rangos con los que se trabajó.

Tabla 12: Graduación de los parámetros de Gradient Boosted Tree

Parámetro	Rango
Iteraciones	[5 ,10, 15, ..., 50]
Profundidad máxima	[3,4,5,6,7,8,9,10]
Tasa de aprendizaje	[0.2, 0.35, ..., 0.8]

Para el algoritmo de *Decision Tree Regressor* se seleccionaron la profundidad máxima de los árboles como también la tasa de aprendizaje, se muestra en la tabla 13 el resumen de los parámetros con sus respectivos rangos.

Tabla 13: Graduación de los parámetros de Decision Tree Regressor

Parámetro	Rango
Profundidad máxima	[3,4,5,6,7,8,9,10]
Tasa de aprendizaje	[0.2, 0.35, ..., 0.8]

A diferencia de los dos algoritmos anteriores *Random Forest* contempla la cantidad de árboles además de la profundidad y la tasa de aprendizaje. La especificación de los rangos utilizados se encuentra en la tabla 14.

Tabla 14: Graduación de parámetros de Random Forest

Parámetro	Rango
Profundidad máxima	[3,4,5,6,7,8,9,10]
Tasa párametroszaje	[0.2, 0.35, ..., 0.8]
Cantidad de árboles	[2, 4, 6, ... , 10]

Para el algoritmo *LinearRegression* se utilizó como único parámetro la cantidad de iteraciones, éstas conformaron un rango de 10 iteraciones por cada prueba, en la tabla 15 se puede apreciar el rango utilizado. Los resultados de para cada iteración fueron pésimos por lo que no se presentará resultados de este algoritmo en las siguientes tablas.

Tabla 15: Graduación de parámetros de *LinearRegression*

Parámetro	Rango
Profundidad máxima	[10,20,30, ... , 100]

Una vez que se construyó la estructura de parámetros a utilizar, se recurrió a la ejecución de las pruebas con variadas combinaciones de éstos, con la finalidad de encontrar el mejor desempeño de cada algoritmo. En la tabla 16 se muestran las distintas pruebas realizadas sobre el segmento automotriz con el algoritmo *Gradient Boosted Tree*.

Tabla 16: Comparación en los resultados de algunas pruebas con *Gradient Boosted Tree*

[Iteraciones, Profundidad, Tasa aprendizaje]	MSE	RMSE	MAE	R^2
[15, 3, 0.5]	39,721	6,302	1,721	0,471
[30, 6, 0.65]	22,531	4,746	1,492	0,772
[50, 10, 0.8]	10,803	3,286	1,3896	0,993

Para ejemplificar la ejecución de *Decision Tree*, se puede apreciar la tabla 17 en donde se encuentra la mejor combinación de sus parámetros para la misma muestra.

Tabla 17: Comparación en los resultados de *Decision Tree Regressor*

[Profundidad, Tasa aprendizaje]	MSE	RMSE	MAE	R^2
[3, 0.5]	39,721	6,302	1,721	0,471
[6, 0.65]	22,531	4,746	1,492	0,772
[10, 0.8]	10,803	3,286	1,3893	0,993

Y finalmente para el algoritmo del *Random Forest* se obtuvieron los resultados expuestos en la tabla 18.

Este mismo proceso se realizó para todos los segmentos de clientes que se describieron al comienzo de esta subfase. Los mejores resultados de cada grupo fueron encontrados en los mismos rangos de parámetros, es por esta razón que los valores para cada grupo de clientes es la que se presenta en la tabla 19, al igual que *LinearRegression* el algoritmo *Random Forest* presenta resultados muy malo, por los parámetros de éstos se excluyen en la tabla 19.

Tabla 18: Comparación en los resultados con *Random Forest*

[Cantidad de árboles, Profundidad, Tasa aprendizaje]	MSE	RMSE	MAE	R^2
[15, 3, 0.5]	6671,221	81,677	77,910	0,012
[30, 6, 0.65]	3329,902	57,705	45,121	0,041
[50, 10, 0.8]	1528,621	39,097	27,285	0,062

Tabla 19: Selección de los parámetros que presentan los mejores resultados para cada algoritmo

	Iteraciones	Profundidad máxima	Tasa de aprendizaje	Cantidad de árboles
<i>Gradient Boosted Tree</i>	50	10	0.8	-
<i>Decision Tree Regressor</i>	-	10	0.8	-

Con estos parámetros se construyó el modelo para cada rubro, en la sección de anexos se encuentra la descripción del modelo de Aerolíneas, sólo se incluye éste debido a que la extensión de dichos modelos es muy grande para incluirlos todos en este documento. Por otro lado se especifica en la tabla 20 la importancia de las variables para cada modelo construido.

Tabla 20: Porcentaje de importancia de cada variable

Rubro	Id	Año	Mes
Aerolíneas	47.57%	32.94%	19.47%
Mercado automotriz	43.34%	37.13%	19.51%
Instituciones Bancarias	46.82%	33.53%	19.64%
Empresas de alta demanda	37.36%	34.93%	27.69%
Empresas de baja demanda	63.65%	24.69%	11.65%

Los resultados se presentan a la empresa en formatos de reportes, los cuales fueron definidos por ellos mismos. Éstos fueron realizados con la herramienta *QlikView*, la cual es especial para la visualización de los datos. Por temas de seguridad y resguardo de los clientes de Ki Technology, sólo se pueden mostrar una parte de los reportes. En la imagen 14 se puede apreciar la cantidad de tickets con respecto al año y mes de una determinada empresa. Se cuenta con filtros búsqueda para que se facilite el trabajo de los usuarios.

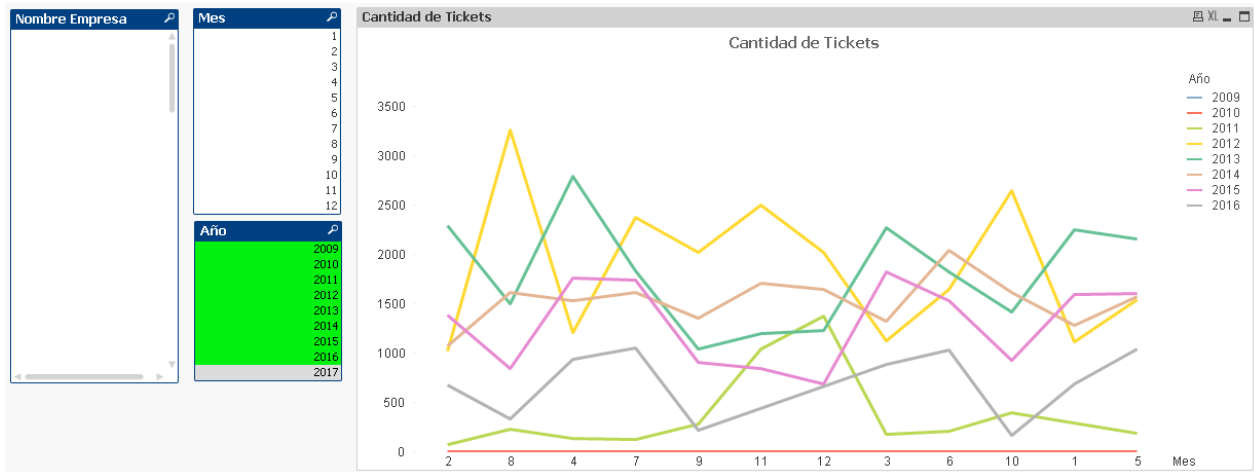


Figura 14: Reportes realizados en QlikView

4.2. Evaluación de los resultados

En esta tarea se realiza la revisión de los resultados obtenidos en la construcción del modelo, para esto se utilizarán las métricas definidas en el plan de pruebas. Se debe destacar que la fiabilidad del modelo está bajo el análisis de los datos usados como datos de entrenamiento.

A continuación se detalla la evaluación de cada segmento de clientes.

1. Instituciones financieras:

Para las instituciones financieras, se probaron distintos algoritmos de minería de datos, con los valores en los parámetros ya indicados en la tabla 19; aquéllos que presentaron un mejor desempeño se presentan en la tabla 21.

Como se puede apreciar los mejores resultados se encuentran en las primeras dos técnicas, *Gradient Boosted Tree* y *Decision Tree Regressor*, donde los valores son idénticos. En ambos casos se puede ver como la suma del error cuadrático medio se reduce considerablemente con respecto al error cuadrático medio otorgado por la técnica *Random Forest*. Se debe recordar que uno de los objetivos de la regresión es poder disminuir al máximo la suma de la diferencia de los residuos. Por otro lado, el coeficiente de correlación es bastante bueno y supera con creces el 70% de aceptación que se requiere para que los resultados presenten datos confiables. El coeficiente de correlación encontrado, indica que la variable dependiente es explicada en un 99% de las variables independientes.

Por último, uno de los aspectos a destacar es la gran exactitud del pronóstico que alcanza este modelo, ya que el error absoluto medio corresponde a tan solo 0,7115, lo que indica que los

valores pronosticados difieren en promedio dicha cantidad. Lo que se busca generalmente que este valor sea lo mínimo posible. Se decide finalmente la utilización del modelo construido con la técnica de *Decision Tree Regressor*, ya que ésta posee un tiempo de procesamiento menor que el empleado por *Gradient Boosted Tree*. Como los resultados del *Random Forest* son muy malos sólo se incluirán en la tabla 20, en las tablas posteriores sólo se presentarán los resultados de *Decision Tree Regressor* y *Gradient Boosted Tree*. En la tabla 22, se presenta una muestra de los valores predichos.

Tabla 21: Resumen de métricas Instituciones financieras

Algoritmo	MSE	RMSE	MAE	R^2
<i>Gradient Boosted Tree</i>	3,090	1,758	0,7115	0,9992
<i>Decision Tree Regressor</i>	3,090	1,758	0,7115	0,9992
<i>Random Forest</i>	3113,85	55,801	36,404	0,230

Tabla 22: Muestra de resultados de Instituciones financieras con *Decision Tree Regressor*

Variables Independientes (Id, Año, Mes)	Valor a predecir	Predicción de la cantidad de <i>tickets</i>	Exactitud
(3, 2012, 7)	4	4,5	88,88%
(5, 2012, 3)	183	183	100%
(120, 2014, 6)	25	23,75	95%
(120, 2105, 6)	23	29	79,31%
(120, 2016, 4)	41	41	100%
(93, 2010, 10)	28	28	100%

2. Aerolíneas:

Como con las instituciones financieras las técnicas que se adaptaron de mejor manera a la dispersión de los datos fueron *Gradient Boosted Tree* y *Decision Tree Regressor*. El resumen de los resultados se presenta en tabla 23.

Como se puede apreciar en la tabla 23, el error cuadrático medio es igual para las primeras dos técnicas. Como el objetivo de la regresión es la disminución de este error, las dos primeras técnicas presentan un rendimiento considerablemente buenas. Por último, el error absoluto medio presenta un valor cercano a cero, lo que muestra la gran exactitud que presenta los modelos de

Decision Tree Regressor y *Gradient Boosted Tree*. La tabla 24 muestra algunas de las predicciones realizadas para este segmento de clientes.

Tabla 23: Resumen de métricas Aerolíneas

Algoritmo	MSE	RMSE	R^2	MAE
<i>Gradient Boosted Tree</i>	7,286	2,699	0,9995	0,764
<i>Decision Tree Regressor</i>	7,286	2,699	0,9995	0,764

Tabla 24: Muestra de resultados de Aerolíneas con *Decision Tree Regressor*

Variables Independientes (Id, Año, Mes)	Valor a predecir	Predicción de la cantidad de tickets	Exactitud
(3, 2012, 7)	15	15	100%
(5, 2012, 3)	43	55.5	77,47%
(120, 2014, 6)	197	196	99,49%
(120, 2015, 6)	53	53	100%
(120, 2016, 4)	200	196	98%
(93, 2010, 10)	7	7	100%

Al igual que en el segmento de instituciones financieras, las técnicas de *Decision Tree Regressor* y *Gradient Boosted Tree* presentan los mismos resultados, pero es la primera la que presenta un mejor tiempo de procesamiento, por esto es que se decidió construir el modelo a partir de esta técnica.

3. Mercado automotriz:

Los mejores resultados se encontraron con las técnicas detalladas en la tabla 25.

Tabla 25: Resumen de métricas Mercado automotriz

Algoritmos	MSE	RMSE	MAE	R^2
<i>Gradient Boosted Tree</i>	10,803	3,286	1,3896	0,9933
<i>Decision Tree Regressor</i>	10,803	3,286	1,3896	0,9933

Para el mercado automotriz, se tiene un aumento error cuadrático medio a diferencia de los encontrados para los dos segmentos anteriores, sin embargo, este segmento de clientes posee una

demanda más grande que las anteriores. Como es la tónica hasta este momento *Decision Tree Regressor* y *Gradient Boosted Tree* presentan las mismas predicciones y por ende los mismos valores en las métricas de evaluación, el coeficiente de correlación supera de manera significativa el umbral de aceptación impuesta en los objetivos de minería de datos. Esto indica que las variables independientes caracterizan en un 99% los valores de la demanda. Por otro lado, el error absoluto medio es un valor bastante pequeño, con lo que la exactitud es bastante buena. Se presenta en la tabla 26 una muestra de las predicciones realizadas bajo la técnica de *Decision Tree Regressor* para este segmento.

Tabla 26: Muestra de resultados del Mercado automotriz con *Decision Tree Regressor*

Variables Independientes (Id, Año, Mes)	Valor a predecir	Predicción de la cantidad de <i>tickets</i>	Exactitud
(16, 2014, 1)	35	35	100%
(72, 2013, 2)	33	26	78,78%
(17, 2012, 10)	26	21,5	82,69%
(20, 2014,4)	96	96	100%
(27, 2013, 2)	36	33	91,6%
(26, 2014, 2)	32	32	100%

Nuevamente se escoge la técnica de *Decision Tree Regressor* por tener los mejores tiempos de procesamiento, en desmedro de *Gradient Boosted Tree*.

4. Empresas de alta demanda:

Este segmento está representado por empresas que demandan de forma continua y que además tienen un comportamiento parecido entre ellas. Las técnicas que presentaron los mejores resultados se encuentran en la tabla 27.

A diferencia de los resultados anteriores, estos son distintos para cada técnica, siendo la mejor *Gradient Boosted Tree*. A su vez éste presenta un muy buen coeficiente de correlación de un 99,73%, el cual supera al estipulado en los objetivos. La exactitud de este modelo se ve representado con el error absoluto medio, el cual es cercano a 2, lo que sigue siendo bajo para la naturaleza de estos datos. Se presenta una muestra de las predicciones realizadas por este modelo en la tabla 28.

Para este caso se selecciona el *Gradient Boosted Tree*, a pesar de que su procesamiento sea más lento que el de *Decision Tree Regressor*, pero se prefiere exactitud en los resultados a menor coste de procesamiento.

Tabla 27: Resumen de métricas Empresas alta demanda

Algoritmos	MSE	RMSE	MAE	R^2
<i>Gradient Boosted Tree</i>	39,339	6,272	1,9482	0,9973
<i>Decision Tree Regressor</i>	60,731	7,793	2,321	0,9959

Tabla 28: Muestra de resultados de Empresas de alta demanda con *Gradient Boosted Tree*

Variables Independientes (Id, Año, Mes)	Valor a predecir	Predicción de la cantidad de <i>tickets</i>	Exactitud
(37, 2012, 12)	343	343	100%
(7, 2012, 5)	351	351	100%
(85, 2014, 2)	78	71,5	91,6%
(98, 2015, 5)	55	107,5	51,16%
(79, 2013, 12)	85	85	100%
(78, 2015, 9)	42	42	100%

5. Empresas de baja demanda:

Por último, se encuentra el segmento de empresas que no compone una demanda demasiado importante, y que presentan algún grado de similitud en la dispersión de sus datos. Los mejores resultados son presentados en la tabla 29.

Tabla 29: Resumen de métricas Empresas de baja demanda

Algoritmos	MSE	RMSE	MAE	R^2
<i>Gradient Boosted Tree</i>	35,3950	5,9493	3,5295	0,7400
<i>Decision Tree Regressor</i>	35,3950	5,9493	3,5295	0,7400

Los resultados son un tanto distinto a lo que se venía repitiendo en los otros modelos, si bien las técnicas que mejores resultados tienen son *Decision Tree Regressor* y *Gradient Boosted Tree*,

éstas no tienen tan buenos resultados como antes. El error cuadrático medio es bastante alto, para la naturaleza de los datos de este segmento de clientes. El mejor coeficiente de correlación es de un 74%, un punto por debajo del establecido como umbral de aceptación. El error absoluto medio es el más alto de todos los modelos, y se considera bastante grande para la naturaleza de los datos de este segmento. Se presenta las predicciones realizadas por este modelo en la tabla 30.

Como los resultados para *Gradient Boosted Tree* y *Decision Tree Regressor* son idénticos, se decide la utilización del segundo debido al tiempo de procesamiento de este.

Como se puede apreciar los resultados de los modelos obtenidos van estrictamente relacionados con la calidad de los datos, ya que para el último modelo la dispersión de los datos complica mucho la elaboración de un modelo con mayor precisión. Es por esto que la elección de los segmentos si bien se hicieron por rubro, también se realizó por el comportamiento de la demanda de éstos a través del tiempo.

Tabla 30: Muestra de resultados de Empresas de baja demanda con *Decision Tree Regressor*

Variables Independientes (Id, Año, Mes)	Valor a predecir	Predicción de la cantidad de <i>tickets</i>	Exactitud
(11, 2011, 6)	13	5,14	39,5%
(15, 2012, 11)	8	8	100%
(18, 2012, 5)	4	5,81	68,84%
(36, 2012, 2)	36	19,3	53,61%
(36, 2012, 4)	53	53	100%
(40, 2011, 12)	13	9,4	72,3%

4.3. Proceso de revisión

Esta etapa corresponde a una evaluación completa de la metodología de CRISP-DM, con el objetivo de identificar las posibles falencias que se pudieran presentar en el transcurso de este trabajo, y de esta manera poder presentar algunas soluciones o mejoras.

El trabajo comienza con la elección de la metodología CRISP-DM, dado que las características de este trabajo se ajustaban bastante bien. Posterior a esto se decidió que la infraestructura de almacenamiento y de procesamiento debía ser del tipo *big data*, ya que la cantidad de los datos que se almacenarán aumentaría de forma significativa en el transcurso del tiempo. La elección

del *framework* de *big data* a utilizar se decidió en base a los principios de innovación; es por esto que la elección fue *Apache Spark*, la que resulta ser la herramienta más reciente para el procesamiento de *big data* por parte de *Apache*. La elección de esta herramienta, en desmedro de *Hadoop*, se sustenta en las características de procesamiento en memoria y la flexibilidad de realizar las aplicaciones en variados ambientes de programación. *Spark* resultó ser un gran acierto, ya que la simplicidad de programar aplicaciones y la bien documentada biblioteca de *machine learning* son ideales para personas que se están iniciando en el mundo del *big data*. Para el procesamiento de los datos, solo se necesitó de la creación de un *cluster* el cual contiene un nodo maestro y un nodo esclavo, éste último albergado en una máquina virtual.

Posterior a la elección de la herramienta de procesamiento, se realizó la extracción de los datos desde las fuentes de datos que proporcionó Ki Teknology. El proceso de ETL fue un elemento que requiere mejoras, ya que no se utilizó ninguna herramienta externa para llevar a cabo esta labor. La extracción y la limpieza de los datos se realizaron mediante *scripts* en Python, los cuales dieron un formato apropiado para la utilización de éstos en el ambiente de *Spark*. La utilización de alguna herramienta de ETL puede ser una gran mejora para este trabajo, ya que una modificación en los datos provocaría que los *scripts* no funcionasen correctamente.

Se construyó un *data mart* el cual permite el almacenamiento de los datos. Creado en un motor de base de datos *Mysql*, *Spark* permite la comunicación mediante *SparkSQL* sin necesidad de una herramienta más sofisticada que la mencionada. Con el almacenamiento de los datos en el formato que corresponde, ya se pudo dar comienzo a la utilización de técnicas de minería de datos para la creación del modelo final.

Para el procesamiento de los datos, se necesitaba una herramienta capaz de minar los datos, tales como *R* o *Weka*. El problema de estas herramientas es que son incompatibles con el procesamiento de *Spark*. Sin embargo *Spark* provee su propia biblioteca de *machine learning*, lo que permite que minar de forma directa y no mediante un tercero. Es por esta razón y por la simplicidad en su utilización que se elige *MLlib* para procesar los datos. Se debe destacar que la cantidad de algoritmos tanto para regresión como para clasificación son abundantes, por lo que se dispone de una amplia gama de opciones para la resolución de problemas.

Las técnicas de minería de datos escogidas fueron las del tipo de regresión, ya que éstas se adaptan a la predicción de variables continuas, como es el caso necesario para esta memoria. Los algoritmos de regresión escogidos son *Linear Regression*, *Decision Tree Regressor*, *Random Forest* y *Gradient Boosted Tree*, siendo los últimos tres los que presentaron los mejores resultados para los modelos construidos

Finalmente se realiza una selección de un *software* capaz de representar de forma legible los resultados, con el objetivo de que la visualización de éstos pueda colaborar en la toma de decisiones. Para esto la herramienta escogida es *QlikView* en su versión de escritorio. Esta herramienta es escogida básicamente por la facilidad con la que se pueden representar los datos y por tener una versión de escritorio gratuita. Se pueden escoger otra herramienta de visualización de datos como *Tableau*, que posee más características que *Qlikview*, pero esto dependerá exclusivamente de Ki Teknology, ya quienes son ellos los que deciden si presupuestar una herramienta que complemente a este trabajo.

4.4. Determinación de futuras fases

Las futuras fases pueden ser mejoras iterativas a los modelos, las cuales vengan como soporte al trabajo ya realizado. Se prevé que para finales del año 2017 se pueda implementar de forma *R*, de forma más permanente en *Apache Spark*, con lo que se provee de una mayor gama de algoritmos de minería de datos, los cuales pueden adaptarse de mejor manera a los datos o incluso obtener los resultados en menor tiempo.

4.5. Plan de Implementación

El plan de implementación cuenta con las instrucciones para la instalación de *Apache Spark*, y la construcción del modelo. La implementación de la infraestructura dependerá de Ki Teknology, ya que ellos son los responsables de destinar servidores físicos para la implementación adecuada de los *cluster*. Se debe señalar que para el desarrollo de este trabajo se utilizaron máquinas virtuales, las que simularon los servidores, es por esto que la configuración de las máquinas virtuales no se detallará.

La base de datos que se utilizará debe permanecer constante en el transcurso del tiempo ya que la extracción de los datos se hizo específicamente para esa base de datos. En caso de que se desee cambiar la base de datos se debe configurar nuevamente la extracción, lo que implicaría un cambio en el código, por lo que se recomienda mantener el formato de los datos.

Por último, se sugiere que se refuerce el hábito a los encargados de registrar los *tickets* de trabajo, ya que una buena calidad y una buena cantidad de estos ayudarán a mejorar los resultados del modelo. Por otro lado, la identificación de las empresas no es del todo adecuada, ya que muchas

de estas se encuentran duplicadas, lo que entorpece la limpieza de los datos y el mismo procesamiento de estos.

1. Instalación de *Apache Spark* en un ambiente de Ubuntu

a. Instalación de Java: *Spark*, requiere de la instalación de Java para su correcto funcionamiento, se recomienda instalar cualquier versión de Java 7, ya que Java 8 requiere un tratamiento distinto y poco documentado para la utilización de *Spark*. La instalación se realizó de la siguiente manera:

- i. `sudo apt-add-repository ppa:webupd8team/java`
- ii. `sudo apt-get update`
- iii. `sudo apt-get install Oracle-java7-installer`

b. Instalación de *Scala*: *Apache Spark* está construido en este lenguaje, por lo que es necesario la instalación de este. Cualquier versión es soportada. Para este caso se utilizó la versión 2.10.4, se debe verificar que el enlace se encuentre disponible para la instalación. Instalación de *Scala* en Ubuntu:

- i. `wget http://www.scala-lang.org/files/archive/scala-2.10.4.tgz`
- ii. `sudo mkdir /usr/local/src/scala`
- iii. `sudo tar xvf scala-2.10.4.tgz -C /usr/local/src/scala/`

c. Modificación del archivo `Bashrc`: Se modifica el archivo `.bashrc`, para que las variables de *Scala* puedan funcionar correctamente cuando se instale *Spark*.

- i. `nano .bashrc`
- ii. `export SCALA_HOME=/usr/local/src/scala/scala-2.10.4`
- iii. `export PATH=$SCALA_HOME/bin:$PATH`
- iv. `..bashrc`

d. Instalación de Git

- i. `sudo apt-get install git`

e. Instalación *Apache Spark*: Finalmente se procede con la instalación propiamente tal de *Spark*. Con las configuraciones anteriores se puede instalar cualquier versión, pero se recomienda que la versión venga con una versión pre fabricada de *Hadoop*, sino la instalación puede tardar varias horas.

- i. `wget http://d3kbcqa49mib13.cloudfront.net/spark-2.0.0-bin-hadoop2.4.tgz`

ii. `tar -xzf spark-2.0.0-bin.hadoop2.4.tgz`

2. Implementación de un modelo predictivo

Para implementar el modelo, se debe primero crear el datamart, especificado en la fase de integración de los datos. Luego de la creación de del datamart, se debe ejecutar el archivo ejecutable que contiene el modelo. Los resultados del modelo se guardan en formato texto, el cual solo debe ser cargado a *Qlikview* para la visualización de estos.

4.6. Plan de monitoreo y mantención

El modelo predictivo puede estar sometido a una mantención del tipo anual, ya que como las predicciones generalmente se realizarán de forma anuales, no tiene mucho sentido estar realizando predicciones todos los días. Por esto una vez que transcurra un año, se debe realizar una mantención con los datos recolectados, para mejorar el entrenamiento del algoritmo, para mejorar y actualizar la calidad del modelo.

El monitoreo si se puede realizar de manera mensual, ya que, con las predicciones realizadas, se puede ir corroborando la certeza y calidad de las predicciones bajo la demanda de los clientes. De esta forma se puede ir corrigiendo para las próximas predicciones las situaciones que requieran algún afinamiento.

Se recomienda, realizar un estudio de los datos antes de realizar cualquier modificación a las configuraciones del modelo, ya que, si se quiere incluir alguna variable predictora, se debe analizar la naturaleza de este nuevo dato y la influencia que tiene tanto con la variable a predecir como también con las otras variables independientes. Por último, se recomienda la mantención adecuada de las máquinas que se encargarán del procesamiento de la información, que sin una debida mantención de estas puede causar una baja en el desempeño del modelo.

4.7. Revisión del proyecto

Es necesario repasar los aspectos que se realizaron de manera correcta, incorrecta y cuáles son los aspectos que se pueden mejorar.

1. Aspectos a destacar: La elaboración de una infraestructura de *big data*, es una herramienta de gran utilidad para el futuro de la empresa, ya que aporta un grado de flexibilidad y escalabilidad en los datos. Esto es replicable para cualquier aspecto o proyecto que requiera del almacenamiento y procesamiento de grandes volúmenes de datos. Por otro lado, la iniciativa de Ki Teknology de comenzar a analizar los datos que antes no había sido usada más que para un control y gestión de sus proyectos. Esto marca una necesidad por parte de ellos en la búsqueda de elementos diferenciadores con la competencia. Es por esto que el análisis de los datos para generar información desde recursos no explotados representa uno de los mayores aspectos a destacar.

Por último, la utilización de *Apache Spark*. La utilización de es nueva y novedosa herramienta es de gran relevancia, ya que su utilización es mucho más simple que Hadoop, la cual se puede considerar como la hermana mayor de Spark. La facilidad de realizar las operaciones, las múltiples librerías y la variedad de lenguajes que soporta hace que sea un elemento indispensable para el desarrollo de aplicaciones pensadas en el *big data*. Como se ha mencionado ya en algunas comunidades *Spark*, dejara en el olvido a *Hadoop* y cada vez más se potenciará su uso para el desarrollo de *big data*.

2. Aspectos incorrectos: Si bien la infraestructura es uno de los mejores aspectos que se le puede otorgar a este trabajo, se debe señalar que los datos que se manipularon para la elaboración del modelo, no cumplen con los requisitos para considerarse como *big data*, por ende la infraestructura se está desaprovechando con la utilización de esta aplicación.
3. Aspectos a mejorar: La calidad de los datos debe mejorar. Ya que algunos atributos de los datos no tienen un formato que entregue algo de información. Es por esto que se recomienda un mejor formato para los atributos de los datos esto permitirá que tanto el procesamiento como la limpieza de los datos sea de forma más expedita. Por otra parte con un abanico más amplios de datos se pueden detectar nuevas variables que ayuden a mejorar las predicciones de los modelos construidos.

Otro aspecto a mejorar es las técnicas utilizadas para evaluar por *Apache Spark*. Como se mencionó anteriormente al ser una herramienta nueva aún muchas de las características de evaluación están siendo desarrolladas. Por este motivo se recomienda hacer un exhaustivo estudio de las métricas a evaluar antes de utilizar *Spark*, ya que es muy probable que se requiera otro *software* de estudio estadístico para apoyar las evaluaciones

de las predicciones. Esto se ve reflejado sobre todo en el estudio de modelos de regresión, donde las métricas de evaluación son pocas.

Conclusiones

Luego de estudiar la variedad de metodologías, se considera que CRISP-DM fue la decisión correcta para el desarrollo de este trabajo, ya que el proceso necesitó un levantamiento de requerimientos y una comprensión del negocio que otras técnicas no otorgaban. Si bien una metodología como SEMMA hubiese agilizado el proceso, la estandarización de las etapas propuesta por la metodología utilizada ayudó a un desarrollo organizado y eficaz. La principal ventaja fue la capacidad de combinar los aspectos del negocio con los aspectos técnicos, ya que la utilización de un plan de trabajo abocado sólo a las necesidades de minería de datos ofrece una planificación bastante útil a la hora de reconocer errores y realizar evaluaciones de forma iterativa.

El primer problema que surgió para trabajar es la poca documentación disponible sobre *Apache Spark*. Al ser una herramienta relativamente nueva no se encontró suficiente ayuda en la *web* por lo que la curva de aprendizaje fue lenta, lo que retrasó un poco en el avance del proyecto. Afortunadamente la manipulación de este *framework* se pudo efectuar de forma satisfactoria resolviendo así la primera problemática que se afrontaba, la cual era trabajar con un volumen de datos no convencional. Con la infraestructura de *big data* ya disponible se logró efectuar todos los procesos descritos en los capítulos anteriores.

El *data mart* creado para guardar la información obtenida del modelo se encuentra físicamente en el servidor de pruebas. De esta manera se concreta uno de los anhelos de la empresa, el cual era comenzar a generar contenedores de información que comiencen la evolución en el análisis de ésta. Es así como se genera un cambio cultural en el tratamiento de datos logrando una utilidad diferente a la característica administrativa que poseía antiguamente; con esto se puede maximizar el uso de sus recursos en pro de los beneficios comerciales que estos puedan otorgar. Lo difícil de la creación de este almacén se tuvo a la hora de realizar los procesos de limpieza, ya que *Apache Spark* no poseía herramientas muy útiles para llevar a cabo esto; actualmente ya se dispone de algunas herramientas nuevas que permiten realizar estas operaciones de una manera más sencilla.

Los mejores resultados encontrados para el modelo se vieron con los algoritmos de *Decision Tree Regressor* y de *Gradient Boosted Tree*, pues son los que mejor se ajustan a este problema en particular. Por otra parte el peor resultado encontrado fue con el algoritmo *Linear Regression* el cual por su naturaleza lineal no se podía ajustar de buena manera al comportamiento oscilatorio que se presentaba recurrentemente. En un primer acercamiento en el análisis de resultados; se

encontró que todos los algoritmos otorgaban muy malos resultados, esto ocurría debido a que no se segmentaron los datos por tipo de clientes. Una vez realizada la segmentación se pudo observar que los resultados fueron mejorando paulatinamente a medida que los parámetros se iban ajustando. Finalmente con el afinamiento de los parámetros los niveles de aceptación preestablecidos fueron superados ampliamente por parte de *Decision Tree Regressor* y de *Gradient Boosted Tree*, en cambio *Random Forest* tuvo un desempeño paupérrimo. Se alcanza la meta de escoger a la mejor familia de algoritmos que resuelven la problemática expuesta para este trabajo.

La definición de las variables críticas fue un proceso que conllevó bastante trabajo, ya que el estudio de éstas sobre la influencia de los resultados no fue sencilla. La importancia de cada una puede ser mejorada agregando los pesos, lo cual no se logró realizar debido a que las pocas herramientas de análisis disponibles no eran lo suficientemente buenas para desarrollar un estudio más exhaustivo. Esta limitante se solucionará con las siguientes versiones de *Apache Spark* en donde se promete una mayor cobertura para este tema. Sin embargo se cumplió con lo requerido debido a que las variables cuyas correlaciones son altas, alcanzan muy buenos niveles de aceptación en los resultados.

Para la realización de las pruebas se crearon dos *data sets*, uno con el cual se entrenó a los datos y otro con el que se realizaron las predicciones. Su confección se basó en una división de 90% de datos que iban desde los años 2009 a 2015 como entrenamiento, mientras que un 10% que abarca una colección desde finales del 2015 a principios del 2016 se utilizó para pruebas, se ocupó dicha segmentación en base a lo establecido por la metodología CRISP-DM. Las predicciones efectuadas sobre los datos de pruebas se comparaban con la demanda real que se tenía, de esta forma a priori se puede visualizar si lo entregado por los algoritmos tiene alguna coherencia con la realidad. Es así como se cumple con un ambiente de entrenamiento y prueba propuesto en las actividades del capítulo 3.

Un aspecto a mejorar en futuras versiones de este proyecto es la automatización en la generación de reportes. Si bien permiten una lectura ágil de los resultados requieren un trabajo extra para su elaboración. El lograr que éstos se creen de forma automática al momento de obtener los resultados del modelo comprendería una herramienta aún más completa. Por esto se insta a las próximas versiones a indagar en sobre este punto.

Las sugerencias que deja este trabajo sobre el tratamiento la manipulación y la administración de los datos, es que se debe generar una estructura que permita explotar y minar los datos de una manera más sencilla. No existe una estandarización en el ingreso de algunos campos que pudiesen ser de gran utilidad para la comprensión de las necesidades de los clientes. Un ejemplo claro es

la descripción de la petición de un cliente, la cual genera un empleado de Ki Teknology puede ser utilizada para aprender no sólo cuando solicitan una acción, sino también que es lo que están solicitando. Hoy en día este campo no cumple con ningún estándar y se limita sólo a ser una gran cadena de caracteres que no aporta en nada a la solución del problema. Muchos campos sufrían el mismo destino, lo cual es una pérdida que debe de ser corregida. Para los futuros proyectos que tenga la empresa, tanto en minería de datos como en otra área, se deben de fijar más en la definición de los datos, pensando que en un futuro ésta puede ser utilizada para generar nuevas oportunidades de negocio.

Por último, sólo se quiere decir que se espera que este trabajo logre en primera instancia ayudar a ser un comienzo en la utilización del aprendizaje automático y de herramientas de minería de datos por parte de Ki Teknology, con la ilusión de que sigan construyendo mejores soluciones tecnológicas para sus clientes y continúen su crecimiento como organización. A su vez se refuerza la idea de seguir complementando esta memoria, tanto por parte de la empresa como también por parte de futuros memoristas, en pos de mejorar los resultados con la utilización de nuevas herramientas que puedan salir en el futuro. Se espera motivar a futuras generaciones de estudiantes en el área de *machine learning*, la cual tiene un campo enorme por explotar y que actualmente no está teniendo toda la atención que debiese de tener. Se insta a todos los interesados a adentrarse en el mundo del *data science* ya que todos los días surgen nuevas e interesantes herramientas que pueden ayudarlos a solucionar más de algún problema.

Finalmente se debe destacar que todo lo realizado solo pudo ser llevado a cabo gracias a todos los conocimientos adquiridos a lo largo de todos estos años de estudios. Tanto los cursos obligatorios como los cursos optativos ayudaron en gran medida a generar esta memoria. Es así como el desarrollo técnico y el análisis hecho se debe en su gran medida a la formación otorgada por la carrera Ingeniería Civil Informática.

Bibliografía

- [1] «Análisis de RFM», [En línea]. Available: https://www.ibm.com/support/knowledgecenter/es/SSLVMB_22.0.0/com.ibm.spss.statistics.help/spss/rfm/rfm_intro.xml.htm. [Último acceso: Abril 2017].
- [2] «Componentes de la metodología CRISP-DM», [En línea]. Available: <https://inteligenciamik.wikispaces.com/COMPONENTES+CRISP-DM>. [Último acceso: 11/11/2016].
- [3] K. M. Han J, «Data Mining: Concepts and Techniques», 2001, [En línea]. Available: <http://www.webmining.cl/2011/01/proceso-de-extraccion-de-conocimiento/>. [Último acceso: Enero 2017].
- [4] Bulkley, «Stages of the SEMMA methodology», 1999, [En línea]. Available: https://www.researchgate.net/figure/271763297_fig7_Figure-11-Stages-of-the-SEMMA-methodology-Bulkley-et-al-1999. [Último acceso: Enero 2017].
- [5] B. Deshpande, «Building the regression model», 11 Noviembre 2014, [En línea]. Available: <http://www.simafore.com/blog/critical-model-steps-for-practical-multiple-linear-regression-pt-2>. [Último acceso: Enero 2017].
- [6] «¿Qué es Big Data?», 2012, [En línea]. Available: <https://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/>. [Último acceso: Diciembre 2016].
- [7] «¿Qué es el Apache Hadoop?», 4 Octubre 2014, [En línea]. Available: <http://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/397377/Qu-es-el-Apache-Hadoop>. [Último acceso: Enero 2017].
- [8] Wikipedia, «Hadoop», Octubre 2016, [En línea]. Available: <https://es.wikipedia.org/w/index.php?title=Hadoop&oldid=94654177>. [Último acceso: Enero 2017].
- [9] S. Yegulalp, «10 ways to query Hadoop with SQL», 16 Septiembre 2014, [En línea]. Available: <http://www.infoworld.com/article/2683729/hadoop/10-ways-to-query-hadoop-with-sql.html>. [Último acceso: Febrero 2017].
- [10] «Big data de código abierto», 2013, [En línea]. Available: Big data de código abierto. [Último acceso: 2017 Enero].
- [11] «Distributed File Systems», 2011, [En línea]. Available: <https://cvw.cac.cornell.edu/mapreduce/dfs?AspxAutoDetectCookieSupport=1>. [Último acceso: Enero 2017].
- [12] «Descripción general de la pasarela de Apache Knox», 2013, [En línea]. Available: https://www.ibm.com/support/knowledgecenter/es/SSPT3X_4.1.0/com.ibm.swg.im.infosphere.biginsights.admin.doc/doc/knox_overview.html. [Último acceso: Enero 2017].
- [13] «What is ZooKeeper?», 2014, [En línea]. Available: <https://www-01.ibm.com/software/data/infosphere/hadoop/zookeeper/>. [Último acceso: 2017 Enero].
- [14] M. P. ESTESO, «APACHE SPARK: QUÉ ES Y CÓMO FUNCIONA», [En línea]. Available: <https://geekytheory.com/apache-spark-que-es-y-como-funciona/>. [Último acceso: Abril 2017].
- [15] J. Vaughan, «Spark vs. Hadoop: ¿es el motor de big data una parte de reemplazo?», TechTarget, 22 Junio 2015, [En línea]. Available: <http://searchdatacenter.techtarget.com/es/cronica/Spark-vs-Hadoop-es-el-motor-de-big-data-una-parte-de-reemplazo>. [Último acceso: 2016 Octubre 24].

- [16] J. Martinez, «Procesos de la inteligencia de negocios», 2002, [En línea]. Available: <http://www.gestiopolis.com/procesos-inteligencia-negocios/>. [Último acceso: Noviembre 2016].

Anexos

Modelo Aerolíneas

```

id in {48,34}
|   id in {48,34,135,47}
|   |   ano in {2011,2012}
|   |   |   mes in {11,12,1,2,3,4,5}
|   |   |   |   ano in {2011,2012,2013,2014}: 9.812 {}
|   |   |   |   ano not in {2011,2012,2013,2014}: 7.983 {}
|   |   |   mes not in {11,12,1,2,3,4,5}
|   |   |   |   ano in {2011,2012,2013,2014,... (3 more)}: 8.195 {}
|   |   |   |   ano not in {2011,2012,2013,2014,... (3 more)}: 7.317
|   |   |   |   ano not in {2011,2012}
|   |   |   |   mes in {11,12,1,2,3,4,5,6}: 8.549 {}
|   |   |   |   mes not in {11,12,1,2,3,4,5,6}
|   |   |   |   id in {48,34,135,47,118}: 7.165 {}
|   |   |   |   id not in {48,34,135,47,118}: 4.965 {}
|   id not in {48,34,135,47}
|   |   mes in {11,12,1,2}
|   |   |   ano in {2011,2012}
|   |   |   |   ano in {2011,2012}: 6.093 {}
|   |   |   |   ano not in {2011,2012}: 0.000 {}
|   |   |   |   ano not in {2011,2012}: 0.000 {}
|   |   |   mes not in {11,12,1,2}
|   |   |   |   mes in {11,12}: -1.556 {}
|   |   |   |   mes not in {11,12}: 0.535 {}
id not in {48,34}
|   ano in {2011,2012}
|   |   id in {48}
|   |   |   ano in {2011,2012}
|   |   |   |   mes in {11,12,1,2,3}: 7.128 {}
|   |   |   |   mes not in {11,12,1,2,3}: 4.525 {}
|   |   |   ano not in {2011,2012}
|   |   |   |   mes in {11,12,1,2,3}: 4.207 {}
|   |   |   |   mes not in {11,12,1,2,3}: 10.845 {}
|   |   id not in {48}
|   |   |   mes in {11,12,1,2,3,4}: 5.811 {}
|   |   |   mes not in {11,12,1,2,3,4}
|   |   |   |   ano in {2011,2012,2013}: 8.244 {}
|   |   |   |   ano not in {2011,2012,2013}: 11.169 {}
|   ano not in {2011,2012}
|   |   mes in {11,12,1}
|   |   |   ano in {2011,2012}: 7.352 {}
|   |   |   ano not in {2011,2012}: 18.885 {}
|   |   mes not in {11,12,1}: 23.227 {}
id in {48,34}
|   id in {48,34,135,47}
|   |   ano in {2011,2012}
|   |   |   mes in {11,12,1,2,3,4,5}
|   |   |   |   ano in {2011,2012,2013,2014}: 9.812 {}
|   |   |   |   ano not in {2011,2012,2013,2014}: 7.983 {}
|   |   |   mes not in {11,12,1,2,3,4,5}
|   |   |   |   ano in {2011,2012,2013,2014,... (3 more)}: 8.195 {}
|   |   |   |   ano not in {2011,2012,2013,2014,... (3 more)}: 7.317
|   |   |   |   ano not in {2011,2012}
|   |   |   |   mes in {11,12,1,2,3,4,5,6}: 8.549 {}
|   |   |   |   mes not in {11,12,1,2,3,4,5,6}

```

```

|   |   |   |   id in {48,34,135,47,118}: 7.165 {}
|   |   |   |   id not in {48,34,135,47,118}: 4.965 {}
|   id not in {48,34,135,47}
|   |   mes in {11,12,1,2}
|   |   |   ano in {2011,2012}
|   |   |   |   ano in {2011,2012}: 6.093 {}
|   |   |   |   ano not in {2011,2012}: 0.000 {}
|   |   |   |   ano not in {2011,2012}: 0.000 {}
|   |   mes not in {11,12,1,2}
|   |   |   mes in {11,12}: 1.556 {}
|   |   |   mes not in {11,12}: 0.535 {}
id not in {48,34}
|   ano in {2011,2012}
|   |   id in {48}
|   |   |   ano in {2011,2012}
|   |   |   |   mes in {11,12,1,2,3}: 7.128 {}
|   |   |   |   mes not in {11,12,1,2,3}: 4.525 {}
|   |   |   |   ano not in {2011,2012}
|   |   |   |   mes in {11,12,1,2,3}: 4.207 {}
|   |   |   |   mes not in {11,12,1,2,3}: 10.845 {}
|   |   id not in {48}
|   |   |   mes in {11,12,1,2,3,4}: 5.811 {}
|   |   |   mes not in {11,12,1,2,3,4}
|   |   |   |   ano in {2011,2012,2013}: 8.244 {}
|   |   |   |   ano not in {2011,2012,2013}: 11.169 {}
|   ano not in {2011,2012}
|   |   mes in {11,12,1}
|   |   |   ano in {2011,2012}: 7.352 {}
|   |   |   ano not in {2011,2012}: 18.885 {}
|   |   mes not in {11,12,1}: 23.227 {}
id in {48,34}
|   id in {48,34,135,47}
|   |   ano in {2011,2012}
|   |   |   mes in {11,12,1,2,3,4}
|   |   |   |   ano in {2011,2012,2013,2014}: 7.704 {}
|   |   |   |   ano not in {2011,2012,2013,2014}: 6.544 {}
|   |   |   mes not in {11,12,1,2,3,4}
|   |   |   |   id in {48,34,135,47,118}: 5.586 {}
|   |   |   |   id not in {48,34,135,47,118}: 6.550 {}
|   |   |   ano not in {2011,2012}
|   |   |   mes in {11,12,1,2,3,4,5,6}: 6.990 {}
|   |   |   mes not in {11,12,1,2,3,4,5,6}
|   |   |   |   id in {48,34,135,47,118}: 5.792 {}
|   |   |   |   id not in {48,34,135,47,118}: 3.978 {}
|   id not in {48,34,135,47}
|   |   mes in {11,12,1,2}
|   |   |   ano in {2011,2012}
|   |   |   |   ano in {2011,2012}: 4.935 {}
|   |   |   |   ano not in {2011,2012}: 0.000 {}
|   |   |   |   ano not in {2011,2012}: 0.000 {}
|   |   mes not in {11,12,1,2}
|   |   |   mes in {11,12}: 1.261 {}
|   |   |   mes not in {11,12}: 0.434 {}
id not in {48,34}
|   ano in {2011,2012}
|   |   id in {48}
|   |   |   ano in {2011,2012}
|   |   |   |   mes in {11,12,1,2,3}: 5.774 {}
|   |   |   |   mes not in {11,12,1,2,3}: 3.665 {}
|   |   |   ano not in {2011,2012}
|   |   |   |   mes in {11,12,1,2,3}: 3.407 {}

```

```

| | | | mes not in {11,12,1,2,3}: 8.785 {}
| | | id not in {48}
| | | | mes in {11,12,1}
| | | | | mes in {11,12,1,2,3,4}: 4.056 {}
| | | | | mes not in {11,12,1,2,3,4}: 6.184 {}
| | | | mes not in {11,12,1}: 9.083 {}
| | ano not in {2011,2012}
| | | mes in {11,12,1}
| | | | ano in {2011,2012}: 5.955 {}
| | | | ano not in {2011,2012}: 15.297 {}
| | | mes not in {11,12,1}: 18.814 {}
id in {48,34}
| | id in {48,34,135,47}
| | | ano in {2011,2012}
| | | | mes in {11,12,1,2,3,4}
| | | | | ano in {2011,2012,2013,2014}: 7.704 {}
| | | | | ano not in {2011,2012,2013,2014}: 6.544 {}
| | | | mes not in {11,12,1,2,3,4}
| | | | | id in {48,34,135,47,118}: 5.586 {}
| | | | | id not in {48,34,135,47,118}: 6.550 {}
| | | | ano not in {2011,2012}
| | | | | mes in {11,12,1,2,3,4,5,6}: 6.990 {}
| | | | | mes not in {11,12,1,2,3,4,5,6}
| | | | | id in {48,34,135,47,118}: 5.792 {}
| | | | | id not in {48,34,135,47,118}: 3.978 {}
| | id not in {48,34,135,47}
| | | mes in {11,12,1,2}
| | | | ano in {2011,2012}
| | | | | ano in {2011,2012}: 4.935 {}
| | | | | ano not in {2011,2012}: 0.000 {}
| | | | | ano not in {2011,2012}: 0.000 {}
| | | mes not in {11,12,1,2}
| | | | mes in {11,12}: 1.261 {}
| | | | mes not in {11,12}: 0.434 {}
id not in {48,34}
| | ano in {2011,2012}
| | | id in {48}
| | | | ano in {2011,2012}
| | | | | mes in {11,12,1,2,3}: 5.774 {}
| | | | | mes not in {11,12,1,2,3}: 3.665 {}
| | | | | ano not in {2011,2012}
| | | | | mes in {11,12,1,2,3}: 3.407 {}
| | | | | mes not in {11,12,1,2,3}: 8.785 {}
| | | id not in {48}
| | | | mes in {11,12,1}
| | | | | mes in {11,12,1,2,3,4}: 4.056 {}
| | | | | mes not in {11,12,1,2,3,4}: 6.184 {}
| | | | | mes not in {11,12,1}: 9.083 {}
| | ano not in {2011,2012}
| | | mes in {11,12,1}
| | | | ano in {2011,2012}: 5.955 {}
| | | | ano not in {2011,2012}: 15.297 {}
| | | mes not in {11,12,1}: 18.814 {}
id in {48,34}
| | id in {48,34,135,47}
| | | ano in {2011,2012}
| | | | mes in {11,12,1,2,3,4}
| | | | | ano in {2011,2012,2013,2014}: 7.704 {}
| | | | | ano not in {2011,2012,2013,2014}: 6.544 {}
| | | | mes not in {11,12,1,2,3,4}
| | | | | id in {48,34,135,47,118}: 5.586 {}

```

```

|   |   |   |   id not in {48,34,135,47,118}: 6.550 {}
|   |   |   |   ano not in {2011,2012}
|   |   |   |   mes in {11,12,1,2,3,4,5,6}: 6.990 {}
|   |   |   |   mes not in {11,12,1,2,3,4,5,6}
|   |   |   |   id in {48,34,135,47,118}: 5.792 {}
|   |   |   |   id not in {48,34,135,47,118}: 3.978 {}
|   id not in {48,34,135,47}
|   |   |   |   mes in {11,12,1,2}
|   |   |   |   |   ano in {2011,2012}
|   |   |   |   |   |   ano in {2011,2012}: 4.935 {}
|   |   |   |   |   |   ano not in {2011,2012}: 0.000 {}
|   |   |   |   |   |   ano not in {2011,2012}: 0.000 {}
|   |   |   |   |   mes not in {11,12,1,2}
|   |   |   |   |   mes in {11,12}: 1.261 {}
|   |   |   |   |   mes not in {11,12}: 0.434 {}
|   id not in {48,34}
|   |   |   |   ano in {2011,2012}
|   |   |   |   |   id in {48}
|   |   |   |   |   |   ano in {2011,2012}
|   |   |   |   |   |   |   mes in {11,12,1,2,3}: 5.774 {}
|   |   |   |   |   |   |   mes not in {11,12,1,2,3}: 3.665 {}
|   |   |   |   |   |   |   ano not in {2011,2012}
|   |   |   |   |   |   |   mes in {11,12,1,2,3}: 3.407 {}
|   |   |   |   |   |   |   mes not in {11,12,1,2,3}: 8.785 {}
|   |   |   |   |   id not in {48}
|   |   |   |   |   |   mes in {11,12,1}
|   |   |   |   |   |   |   mes in {11,12,1,2,3,4}: 4.056 {}
|   |   |   |   |   |   |   mes not in {11,12,1,2,3,4}: 6.184 {}
|   |   |   |   |   |   |   mes not in {11,12,1}: 9.083 {}
|   |   |   |   |   ano not in {2011,2012}
|   |   |   |   |   |   mes in {11,12,1}
|   |   |   |   |   |   |   ano in {2011,2012}: 5.955 {}
|   |   |   |   |   |   |   ano not in {2011,2012}: 15.297 {}
|   |   |   |   |   |   mes not in {11,12,1}: 18.814 {}
|   id in {48,34}
|   |   |   |   id in {48,34,135,47}
|   |   |   |   |   ano in {2011,2012}
|   |   |   |   |   |   mes in {11,12,1,2,3,4}
|   |   |   |   |   |   |   ano in {2011,2012,2013,2014}: 7.704 {}
|   |   |   |   |   |   |   ano not in {2011,2012,2013,2014}: 6.544 {}
|   |   |   |   |   |   |   mes not in {11,12,1,2,3,4}
|   |   |   |   |   |   |   id in {48,34,135,47,118}: 5.586 {}
|   |   |   |   |   |   |   id not in {48,34,135,47,118}: 6.550 {}
|   |   |   |   |   |   ano not in {2011,2012}
|   |   |   |   |   |   |   mes in {11,12,1,2,3,4,5,6}: 6.990 {}
|   |   |   |   |   |   |   mes not in {11,12,1,2,3,4,5,6}
|   |   |   |   |   |   |   id in {48,34,135,47,118}: 5.792 {}
|   |   |   |   |   |   |   id not in {48,34,135,47,118}: 3.978 {}
|   |   |   |   |   id not in {48,34,135,47}
|   |   |   |   |   |   mes in {11,12,1,2}
|   |   |   |   |   |   |   ano in {2011,2012}
|   |   |   |   |   |   |   |   ano in {2011,2012}: 4.935 {}
|   |   |   |   |   |   |   |   ano not in {2011,2012}: 0.000 {}
|   |   |   |   |   |   |   |   ano not in {2011,2012}: 0.000 {}
|   |   |   |   |   |   |   mes not in {11,12,1,2}
|   |   |   |   |   |   |   |   mes in {11,12}: 1.261 {}
|   |   |   |   |   |   |   |   mes not in {11,12}: 0.434 {}
|   id not in {48,34}
|   |   |   |   |   ano in {2011,2012}
|   |   |   |   |   |   id in {48}
|   |   |   |   |   |   |   ano in {2011,2012}

```

```

| | | | mes in {11,12,1,2,3}: 5.774 {}
| | | | mes not in {11,12,1,2,3}: 3.665 {}
| | | | ano not in {2011,2012}
| | | | mes in {11,12,1,2,3}: 3.407 {}
| | | | mes not in {11,12,1,2,3}: 8.785 {}
| | | id not in {48}
| | | | mes in {11,12,1}
| | | | | mes in {11,12,1,2,3,4}: 4.056 {}
| | | | | mes not in {11,12,1,2,3,4}: 6.184 {}
| | | | mes not in {11,12,1}: 9.083 {}
| | ano not in {2011,2012}
| | | mes in {11,12,1}
| | | | ano in {2011,2012}: 5.955 {}
| | | | ano not in {2011,2012}: 15.297 {}
| | | mes not in {11,12,1}: 18.814 {}
id in {48,34}
| | id in {48,34,135,47}
| | | ano in {2011,2012}
| | | | mes in {11,12,1,2,3,4}
| | | | | ano in {2011,2012,2013,2014}: 7.704 {}
| | | | | ano not in {2011,2012,2013,2014}: 6.544 {}
| | | | mes not in {11,12,1,2,3,4}
| | | | | id in {48,34,135,47,118}: 5.586 {}
| | | | | id not in {48,34,135,47,118}: 6.550 {}
| | | ano not in {2011,2012}
| | | | mes in {11,12,1,2,3,4,5,6}: 6.990 {}
| | | | mes not in {11,12,1,2,3,4,5,6}
| | | | | id in {48,34,135,47,118}: -5.792 {}
| | | | | id not in {48,34,135,47,118}: 3.978 {}
| | id not in {48,34,135,47}
| | | mes in {11,12,1,2}
| | | | ano in {2011,2012}
| | | | | ano in {2011,2012}: 4.935 {}
| | | | | ano not in {2011,2012}: 0.000 {}
| | | | ano not in {2011,2012}: 0.000 {}
| | | mes not in {11,12,1,2}
| | | | mes in {11,12}: 1.261 {}
| | | | mes not in {11,12}: 0.434 {}
id not in {48,34}
| | ano in {2011,2012}
| | | id in {48}
| | | | ano in {2011,2012}
| | | | | mes in {11,12,1,2,3}: 5.774 {}
| | | | | mes not in {11,12,1,2,3}: 3.665 {}
| | | | ano not in {2011,2012}
| | | | | mes in {11,12,1,2,3}: 3.407 {}
| | | | | mes not in {11,12,1,2,3}: 8.785 {}
| | | id not in {48}
| | | | mes in {11,12,1}
| | | | | mes in {11,12,1,2,3,4}: 4.056 {}
| | | | | mes not in {11,12,1,2,3,4}: 6.184 {}
| | | | mes not in {11,12,1}: 9.083 {}
| | ano not in {2011,2012}
| | | mes in {11,12,1}
| | | | ano in {2011,2012}: 5.955 {}
| | | | ano not in {2011,2012}: 15.297 {}
| | | mes not in {11,12,1}: 18.814 {}
id in {48,34}
| | id in {48,34,135}
| | | mes in {11,12,1}
| | | | ano in {2011}

```

```

|   |   |   |   mes in {11,12,1,2,3,4}: 4.346 {}
|   |   |   |   mes not in {11,12,1,2,3,4}: 3.493 {}
|   |   |   |   ano not in {2011}: -3.204 {}
|   |   |   |   mes not in {11,12,1}
|   |   |   |   id in {48,34,135,47,118}: 2.253 {}
|   |   |   |   id not in {48,34,135,47,118}
|   |   |   |   ano in {2011,2012}: 2.419 {}
|   |   |   |   ano not in {2011,2012}: 2.984 {}
|   |   |   |   id not in {48,34,135}
|   |   |   |   mes in {11,12,1,2,3,4}
|   |   |   |   mes in {11,12,1,2,3,4,5,6,7}: 3.253 {}
|   |   |   |   mes not in {11,12,1,2,3,4,5,6,7}: 2.103 {}
|   |   |   |   mes not in {11,12,1,2,3,4}
|   |   |   |   ano in {2011,2012}: 0.767 {}
|   |   |   |   ano not in {2011,2012}: 0.489 {}
id not in {48,34}
|   |   |   |   ano in {2011}
|   |   |   |   mes in {11,12,1,2,3,4,5}
|   |   |   |   |   ano in {2011,2012}
|   |   |   |   |   |   id in {48}: 5.212 {}
|   |   |   |   |   |   id not in {48}: 0.850 {}
|   |   |   |   |   |   ano not in {2011,2012}: 5.205 {}
|   |   |   |   |   |   mes not in {11,12,1,2,3,4,5}
|   |   |   |   |   |   ano in {2011}
|   |   |   |   |   |   |   ano in {2011}: 2.195 {}
|   |   |   |   |   |   |   ano not in {2011}: 0.000 {}
|   |   |   |   |   |   |   ano not in {2011}: 13.424 {}
|   |   |   |   |   |   |   ano not in {2011}
|   |   |   |   |   |   |   mes in {11,12,1,2,3}: 2.181 {}
|   |   |   |   |   |   |   mes not in {11,12,1,2,3}: 26.215 {}
id in {48,34}
|   |   |   |   id in {48,34,135}
|   |   |   |   |   mes in {11,12,1}
|   |   |   |   |   |   ano in {2011}
|   |   |   |   |   |   |   mes in {11,12,1,2,3,4}: 4.346 {}
|   |   |   |   |   |   |   mes not in {11,12,1,2,3,4}: 3.493 {}
|   |   |   |   |   |   |   ano not in {2011}: -3.204 {}
|   |   |   |   |   |   |   mes not in {11,12,1}
|   |   |   |   |   |   |   id in {48,34,135,47,118}: 2.253 {}
|   |   |   |   |   |   |   id not in {48,34,135,47,118}
|   |   |   |   |   |   |   ano in {2011,2012}: 2.419 {}
|   |   |   |   |   |   |   ano not in {2011,2012}: 2.984 {}
|   |   |   |   |   |   |   id not in {48,34,135}
|   |   |   |   |   |   |   mes in {11,12,1,2,3,4}
|   |   |   |   |   |   |   mes in {11,12,1,2,3,4,5,6,7}: 3.253 {}
|   |   |   |   |   |   |   mes not in {11,12,1,2,3,4,5,6,7}: 2.103 {}
|   |   |   |   |   |   |   mes not in {11,12,1,2,3,4}
|   |   |   |   |   |   |   ano in {2011,2012}: 0.767 {}
|   |   |   |   |   |   |   ano not in {2011,2012}: 0.489 {}
id not in {48,34}
|   |   |   |   ano in {2011}
|   |   |   |   mes in {11,12,1,2,3,4,5}
|   |   |   |   |   ano in {2011,2012}
|   |   |   |   |   |   id in {48}: 5.212 {}
|   |   |   |   |   |   id not in {48}: 0.850 {}
|   |   |   |   |   |   ano not in {2011,2012}: 5.205 {}
|   |   |   |   |   |   mes not in {11,12,1,2,3,4,5}
|   |   |   |   |   |   ano in {2011}
|   |   |   |   |   |   |   ano in {2011}: 2.195 {}
|   |   |   |   |   |   |   ano not in {2011}: 0.000 {}
|   |   |   |   |   |   |   ano not in {2011}: 13.424 {}

```

```

|   ano not in {2011}
|   |   mes in {11,12,1,2,3}: 2.181 {}
|   |   mes not in {11,12,1,2,3}: 26.215 {}
id in {48,34}
|   id in {48,34,135}
|   |   mes in {11,12,1}
|   |   |   ano in {2011}
|   |   |   |   mes in {11,12,1,2,3,4}: 4.346 {}
|   |   |   |   mes not in {11,12,1,2,3,4}: 3.493 {}
|   |   |   ano not in {2011}: -3.204 {}
|   |   mes not in {11,12,1}
|   |   |   id in {48,34,135,47,118}: 2.253 {}
|   |   |   id not in {48,34,135,47,118}
|   |   |   |   ano in {2011,2012}: 2.419 {}
|   |   |   |   ano not in {2011,2012}: 2.984 {}
|   id not in {48,34,135}
|   |   mes in {11,12,1,2,3,4}
|   |   |   mes in {11,12,1,2,3,4,5,6,7}: 3.253 {}
|   |   |   mes not in {11,12,1,2,3,4,5,6,7}: 2.103 {}
|   |   mes not in {11,12,1,2,3,4}
|   |   |   ano in {2011,2012}: 0.767 {}
|   |   |   ano not in {2011,2012}: 0.489 {}
id not in {48,34}
|   ano in {2011}
|   |   mes in {11,12,1,2,3,4,5}
|   |   |   ano in {2011,2012}
|   |   |   |   id in {48}: 5.212 {}
|   |   |   |   id not in {48}: 0.850 {}
|   |   |   ano not in {2011,2012}: 5.205 {}
|   |   mes not in {11,12,1,2,3,4,5}
|   |   |   ano in {2011}
|   |   |   |   ano in {2011}: 2.195 {}
|   |   |   |   ano not in {2011}: 0.000 {}
|   |   |   ano not in {2011}: 13.424 {}
|   ano not in {2011}
|   |   mes in {11,12,1,2,3}: 2.181 {}
|   |   mes not in {11,12,1,2,3}: 26.215 {}
id in {48,34}
|   id in {48,34,135}
|   |   mes in {11,12,1}
|   |   |   ano in {2011}
|   |   |   |   mes in {11,12,1,2,3,4}: 4.346 {}
|   |   |   |   mes not in {11,12,1,2,3,4}: 3.493 {}
|   |   |   ano not in {2011}: -3.204 {}
|   |   mes not in {11,12,1}
|   |   |   id in {48,34,135,47,118}: 2.253 {}
|   |   |   id not in {48,34,135,47,118}
|   |   |   |   ano in {2011,2012}: 2.419 {}
|   |   |   |   ano not in {2011,2012}: 2.984 {}
|   id not in {48,34,135}
|   |   mes in {11,12,1,2,3,4}
|   |   |   mes in {11,12,1,2,3,4,5,6,7}: 3.253 {}
|   |   |   mes not in {11,12,1,2,3,4,5,6,7}: 2.103 {}
|   |   mes not in {11,12,1,2,3,4}
|   |   |   ano in {2011,2012}: 0.767 {}
|   |   |   ano not in {2011,2012}: 0.489 {}
id not in {48,34}
|   ano in {2011}
|   |   mes in {11,12,1,2,3,4,5}
|   |   |   ano in {2011,2012}
|   |   |   |   id in {48}: 5.212 {}

```

```

|   |   |   |   id not in {48}: 0.850 {}
|   |   |   |   ano not in {2011,2012}: 5.205 {}
|   |   |   |   mes not in {11,12,1,2,3,4,5}
|   |   |   |   ano in {2011}
|   |   |   |   |   ano in {2011}: 2.195 {}
|   |   |   |   |   ano not in {2011}: 0.000 {}
|   |   |   |   |   ano not in {2011}: 13.424 {}
|   |   |   |   ano not in {2011}
|   |   |   |   |   mes in {11,12,1,2,3}: 2.181 {}
|   |   |   |   |   mes not in {11,12,1,2,3}: 26.215 {}
id in {48,34}
|   |   |   |   id in {48,34,135}
|   |   |   |   |   mes in {11,12,1}
|   |   |   |   |   |   ano in {2011}
|   |   |   |   |   |   |   mes in {11,12,1,2,3,4}: 4.346 {}
|   |   |   |   |   |   |   mes not in {11,12,1,2,3,4}: 3.493 {}
|   |   |   |   |   |   |   ano not in {2011}: -3.204 {}
|   |   |   |   |   |   |   mes not in {11,12,1}
|   |   |   |   |   |   |   id in {48,34,135,47,118}: 2.253 {}
|   |   |   |   |   |   |   id not in {48,34,135,47,118}
|   |   |   |   |   |   |   |   ano in {2011,2012}: 2.419 {}
|   |   |   |   |   |   |   |   ano not in {2011,2012}: 2.984 {}
|   |   |   |   |   id not in {48,34,135}
|   |   |   |   |   |   mes in {11,12,1,2,3,4}
|   |   |   |   |   |   |   mes in {11,12,1,2,3,4,5,6,7}: 3.253 {}
|   |   |   |   |   |   |   mes not in {11,12,1,2,3,4,5,6,7}: 2.103 {}
|   |   |   |   |   |   |   mes not in {11,12,1,2,3,4}
|   |   |   |   |   |   |   |   ano in {2011,2012}: -0.767 {}
|   |   |   |   |   |   |   |   ano not in {2011,2012}: 0.489 {}
id not in {48,34}
|   |   |   |   |   ano in {2011}
|   |   |   |   |   |   mes in {11,12,1,2,3,4,5}
|   |   |   |   |   |   |   ano in {2011,2012}
|   |   |   |   |   |   |   |   id in {48}: 5.212 {}
|   |   |   |   |   |   |   |   id not in {48}: 0.850 {}
|   |   |   |   |   |   |   |   ano not in {2011,2012}: 5.205 {}
|   |   |   |   |   |   |   |   mes not in {11,12,1,2,3,4,5}
|   |   |   |   |   |   |   |   |   ano in {2011}
|   |   |   |   |   |   |   |   |   |   ano in {2011}: 2.195 {}
|   |   |   |   |   |   |   |   |   |   ano not in {2011}: 0.000 {}
|   |   |   |   |   |   |   |   |   |   ano not in {2011}: 13.424 {}
|   |   |   |   |   |   |   |   ano not in {2011}
|   |   |   |   |   |   |   |   |   |   mes in {11,12,1,2,3}: -2.181 {}
|   |   |   |   |   |   |   |   |   |   mes not in {11,12,1,2,3}: 26.215 {}
id in {48,34}
|   |   |   |   |   id in {48,34,135}
|   |   |   |   |   |   mes in {11,12,1}
|   |   |   |   |   |   |   ano in {2011}
|   |   |   |   |   |   |   |   mes in {11,12,1,2,3,4}: 4.346 {}
|   |   |   |   |   |   |   |   mes not in {11,12,1,2,3,4}: 3.493 {}
|   |   |   |   |   |   |   |   ano not in {2011}: -3.204 {}
|   |   |   |   |   |   |   |   mes not in {11,12,1}
|   |   |   |   |   |   |   |   |   id in {48,34,135,47,118}: 2.253 {}
|   |   |   |   |   |   |   |   |   id not in {48,34,135,47,118}
|   |   |   |   |   |   |   |   |   |   ano in {2011,2012}: 2.419 {}
|   |   |   |   |   |   |   |   |   |   ano not in {2011,2012}: 2.984 {}
|   |   |   |   |   |   |   |   id not in {48,34,135}
|   |   |   |   |   |   |   |   |   mes in {11,12,1,2,3,4}
|   |   |   |   |   |   |   |   |   |   mes in {11,12,1,2,3,4,5,6,7}: 3.253 {}
|   |   |   |   |   |   |   |   |   |   mes not in {11,12,1,2,3,4,5,6,7}: 2.103 {}
|   |   |   |   |   |   |   |   |   |   mes not in {11,12,1,2,3,4}

```

```

|   |   |   ano in {2011,2012}: 0.767 {}
|   |   |   ano not in {2011,2012}: 0.489 {}
id not in {48,34}
|   ano in {2011}
|   |   mes in {11,12,1,2,3,4,5}
|   |   |   ano in {2011,2012}
|   |   |   |   id in {48}: -5.212 {}
|   |   |   |   id not in {48}: 0.850 {}
|   |   |   ano not in {2011,2012}: 5.205 {}
|   |   mes not in {11,12,1,2,3,4,5}
|   |   |   ano in {2011}
|   |   |   |   ano in {2011}: 2.195 {}
|   |   |   |   ano not in {2011}: 0.000 {}
|   |   |   ano not in {2011}: 13.424 {}
|   ano not in {2011}
|   |   mes in {11,12,1,2,3}: 2.181 {}
|   |   mes not in {11,12,1,2,3}: 26.215 {}
id in {48,34}
|   id in {48,34,135}
|   |   mes in {11,12,1}
|   |   |   ano in {2011}
|   |   |   |   mes in {11,12,1,2,3,4}: 4.346 {}
|   |   |   |   mes not in {11,12,1,2,3,4}: 3.493 {}
|   |   |   ano not in {2011}: -3.204 {}
|   |   mes not in {11,12,1}
|   |   |   id in {48,34,135,47,118}: 2.253 {}
|   |   |   id not in {48,34,135,47,118}
|   |   |   |   ano in {2011,2012}: 2.419 {}
|   |   |   |   ano not in {2011,2012}: 2.984 {}
|   id not in {48,34,135}
|   |   mes in {11,12,1,2,3,4}
|   |   |   mes in {11,12,1,2,3,4,5,6,7}: 3.253 {}
|   |   |   mes not in {11,12,1,2,3,4,5,6,7}: 2.103 {}
|   |   mes not in {11,12,1,2,3,4}
|   |   |   ano in {2011,2012}: 0.767 {}
|   |   |   ano not in {2011,2012}: 0.489 {}
id not in {48,34}
|   ano in {2011}
|   |   mes in {11,12,1,2,3,4,5}
|   |   |   ano in {2011,2012}
|   |   |   |   id in {48}: 5.212 {}
|   |   |   |   id not in {48}: 0.850 {}
|   |   |   ano not in {2011,2012}: 5.205 {}
|   |   mes not in {11,12,1,2,3,4,5}
|   |   |   ano in {2011}
|   |   |   |   ano in {2011}: 2.195 {}
|   |   |   |   ano not in {2011}: 0.000 {}
|   |   |   ano not in {2011}: 13.424 {}
|   ano not in {2011}
|   |   mes in {11,12,1,2,3}: 2.181 {}
|   |   mes not in {11,12,1,2,3}: 26.215 {}
id in {48,34}
|   id in {48,34,135}
|   |   mes in {11,12,1}
|   |   |   ano in {2011}
|   |   |   |   mes in {11,12,1,2,3,4}: 4.346 {}
|   |   |   |   mes not in {11,12,1,2,3,4}: 3.493 {}
|   |   |   ano not in {2011}: -3.204 {}
|   |   mes not in {11,12,1}
|   |   |   id in {48,34,135,47,118}: 2.253 {}
|   |   |   id not in {48,34,135,47,118}

```

```

|   |   |   |   ano in {2011,2012}: 2.419 {}
|   |   |   |   ano not in {2011,2012}: 2.984 {}
|   id not in {48,34,135}
|   |   mes in {11,12,1,2,3,4}
|   |   |   mes in {11,12,1,2,3,4,5,6,7}: 3.253 {}
|   |   |   mes not in {11,12,1,2,3,4,5,6,7}: 2.103 {}
|   |   mes not in {11,12,1,2,3,4}
|   |   |   ano in {2011,2012}: 0.767 {}
|   |   |   ano not in {2011,2012}: 0.489 {}
id not in {48,34}
|   ano in {2011}
|   |   mes in {11,12,1,2,3,4,5}
|   |   |   ano in {2011,2012}
|   |   |   |   id in {48}: 5.212 {}
|   |   |   |   id not in {48}: 0.850 {}
|   |   |   ano not in {2011,2012}: 5.205 {}
|   |   mes not in {11,12,1,2,3,4,5}
|   |   |   ano in {2011}
|   |   |   |   ano in {2011}: 2.195 {}
|   |   |   |   ano not in {2011}: 0.000 {}
|   |   |   ano not in {2011}: 13.424 {}
|   ano not in {2011}
|   |   mes in {11,12,1,2,3}: -2.181 {}
|   |   mes not in {11,12,1,2,3}: 26.215 {}
id in {48,34}
|   id in {48,34,135}
|   |   mes in {11,12,1}
|   |   |   ano in {2011}
|   |   |   |   mes in {11,12,1,2,3,4}: 4.346 {}
|   |   |   |   mes not in {11,12,1,2,3,4}: 3.493 {}
|   |   |   ano not in {2011}: -3.204 {}
|   |   mes not in {11,12,1}
|   |   |   id in {48,34,135,47,118}: 2.253 {}
|   |   |   id not in {48,34,135,47,118}
|   |   |   |   ano in {2011,2012}: 2.419 {}
|   |   |   |   ano not in {2011,2012}: 2.984 {}
|   id not in {48,34,135}
|   |   mes in {11,12,1,2,3,4}
|   |   |   mes in {11,12,1,2,3,4,5,6,7}: 3.253 {}
|   |   |   mes not in {11,12,1,2,3,4,5,6,7}: 2.103 {}
|   |   mes not in {11,12,1,2,3,4}
|   |   |   ano in {2011,2012}: 0.767 {}
|   |   |   ano not in {2011,2012}: 0.489 {}
id not in {48,34}
|   ano in {2011}
|   |   mes in {11,12,1,2,3,4,5}
|   |   |   ano in {2011,2012}
|   |   |   |   id in {48}: 5.212 {}
|   |   |   |   id not in {48}: 0.850 {}
|   |   |   ano not in {2011,2012}: 5.205 {}
|   |   mes not in {11,12,1,2,3,4,5}
|   |   |   ano in {2011}
|   |   |   |   ano in {2011}: 2.195 {}
|   |   |   |   ano not in {2011}: 0.000 {}
|   |   |   ano not in {2011}: 13.424 {}
|   ano not in {2011}
|   |   mes in {11,12,1,2,3}: 2.181 {}
|   |   mes not in {11,12,1,2,3}: 26.215 {}
id in {48,34}
|   id in {48,34,135}
|   |   mes in {11,12,1}

```

```

|   |   |   ano in {2011}
|   |   |   |   mes in {11,12,1,2,3,4}: 4.346 {}
|   |   |   |   mes not in {11,12,1,2,3,4}: 3.493 {}
|   |   |   ano not in {2011}: -3.204 {}
|   |   mes not in {11,12,1}
|   |   |   id in {48,34,135,47,118}: 2.253 {}
|   |   |   id not in {48,34,135,47,118}
|   |   |   |   ano in {2011,2012}: 2.419 {}
|   |   |   |   ano not in {2011,2012}: 2.984 {}
|   id not in {48,34,135}
|   |   mes in {11,12,1,2,3,4}
|   |   |   mes in {11,12,1,2,3,4,5,6,7}: 3.253 {}
|   |   |   mes not in {11,12,1,2,3,4,5,6,7}: 2.103 {}
|   |   mes not in {11,12,1,2,3,4}
|   |   |   ano in {2011,2012}: 0.767 {}
|   |   |   ano not in {2011,2012}: 0.489 {}
id not in {48,34}
|   ano in {2011}
|   |   mes in {11,12,1,2,3,4,5}
|   |   |   ano in {2011,2012}
|   |   |   |   id in {48}: -5.212 {}
|   |   |   |   id not in {48}: 0.850 {}
|   |   |   ano not in {2011,2012}: 5.205 {}
|   |   mes not in {11,12,1,2,3,4,5}
|   |   |   ano in {2011}
|   |   |   |   ano in {2011}: 2.195 {}
|   |   |   |   ano not in {2011}: 0.000 {}
|   |   |   ano not in {2011}: 13.424 {}
|   ano not in {2011}
|   |   mes in {11,12,1,2,3}: 2.181 {}
|   |   mes not in {11,12,1,2,3}: 26.215 {}
id in {48,34}
|   id in {48,34,135}
|   |   mes in {11,12,1}
|   |   |   ano in {2011}
|   |   |   |   mes in {11,12,1,2,3,4}: 4.346 {}
|   |   |   |   mes not in {11,12,1,2,3,4}: 3.493 {}
|   |   |   ano not in {2011}: 3.204 {}
|   |   mes not in {11,12,1}
|   |   |   id in {48,34,135,47,118}: 2.253 {}
|   |   |   id not in {48,34,135,47,118}
|   |   |   |   ano in {2011,2012}: 2.419 {}
|   |   |   |   ano not in {2011,2012}: 2.984 {}
|   id not in {48,34,135}
|   |   mes in {11,12,1,2,3,4}
|   |   |   mes in {11,12,1,2,3,4,5,6,7}: 3.253 {}
|   |   |   mes not in {11,12,1,2,3,4,5,6,7}: 2.103 {}
|   |   mes not in {11,12,1,2,3,4}
|   |   |   ano in {2011,2012}: 0.767 {}
|   |   |   ano not in {2011,2012}: 0.489 {}
id not in {48,34}
|   ano in {2011}
|   |   mes in {11,12,1,2,3,4,5}
|   |   |   ano in {2011,2012}
|   |   |   |   id in {48}: 5.212 {}
|   |   |   |   id not in {48}: 0.850 {}
|   |   |   ano not in {2011,2012}: 5.205 {}
|   |   mes not in {11,12,1,2,3,4,5}
|   |   |   ano in {2011}
|   |   |   |   ano in {2011}: 2.195 {}
|   |   |   |   ano not in {2011}: 0.000 {}

```

```

|   |   |   ano not in {2011}: 13.424 {}
|   ano not in {2011}
|   |   mes in {11,12,1,2,3}: 2.181 {}
|   |   mes not in {11,12,1,2,3}: 26.215 {}
id in {48,34}
|   id in {48,34,135}
|   |   mes in {11,12,1}
|   |   |   ano in {2011}
|   |   |   |   mes in {11,12,1,2,3,4}: 4.346 {}
|   |   |   |   mes not in {11,12,1,2,3,4}: 3.493 {}
|   |   |   ano not in {2011}: 3.204 {}
|   |   mes not in {11,12,1}
|   |   |   id in {48,34,135,47,118}: 2.253 {}
|   |   |   id not in {48,34,135,47,118}
|   |   |   |   ano in {2011,2012}: 2.419 {}
|   |   |   |   ano not in {2011,2012}: 2.984 {}
|   id not in {48,34,135}
|   |   mes in {11,12,1,2,3,4}
|   |   |   mes in {11,12,1,2,3,4,5,6,7}: 3.253 {}
|   |   |   mes not in {11,12,1,2,3,4,5,6,7}: 2.103 {}
|   |   mes not in {11,12,1,2,3,4}
|   |   |   ano in {2011,2012}: -0.767 {}
|   |   |   ano not in {2011,2012}: 0.489 {}
id not in {48,34}
|   ano in {2011}
|   |   mes in {11,12,1,2,3,4,5}
|   |   |   ano in {2011,2012}
|   |   |   |   id in {48}: -5.212 {}
|   |   |   |   id not in {48}: 0.850 {}
|   |   |   ano not in {2011,2012}: 5.205 {}
|   |   mes not in {11,12,1,2,3,4,5}
|   |   |   ano in {2011}
|   |   |   |   ano in {2011}: 2.195 {}
|   |   |   |   ano not in {2011}: 0.000 {}
|   |   |   ano not in {2011}: 13.424 {}
|   ano not in {2011}
|   |   mes in {11,12,1,2,3}: 2.181 {}
|   |   mes not in {11,12,1,2,3}: 26.215 {}
id in {48,34}
|   id in {48,34,135}
|   |   mes in {11,12,1}
|   |   |   ano in {2011}
|   |   |   |   mes in {11,12,1,2,3,4}: 4.346 {}
|   |   |   |   mes not in {11,12,1,2,3,4}: 3.493 {}
|   |   |   ano not in {2011}: -3.204 {}
|   |   mes not in {11,12,1}
|   |   |   id in {48,34,135,47,118}: 2.253 {}
|   |   |   id not in {48,34,135,47,118}
|   |   |   |   ano in {2011,2012}: 2.419 {}
|   |   |   |   ano not in {2011,2012}: 2.984 {}
|   id not in {48,34,135}
|   |   mes in {11,12,1,2,3,4}
|   |   |   mes in {11,12,1,2,3,4,5,6,7}: 3.253 {}
|   |   |   mes not in {11,12,1,2,3,4,5,6,7}: 2.103 {}
|   |   mes not in {11,12,1,2,3,4}
|   |   |   ano in {2011,2012}: 0.767 {}
|   |   |   ano not in {2011,2012}: 0.489 {}
id not in {48,34}
|   ano in {2011}
|   |   mes in {11,12,1,2,3,4,5}
|   |   |   ano in {2011,2012}

```

```

|   |   |   |   id in {48}: 5.212 {}
|   |   |   |   id not in {48}: 0.850 {}
|   |   |   ano not in {2011,2012}: 5.205 {}
|   |   mes not in {11,12,1,2,3,4,5}
|   |   |   ano in {2011}
|   |   |   |   ano in {2011}: 2.195 {}
|   |   |   |   ano not in {2011}: 0.000 {}
|   |   |   ano not in {2011}: 13.424 {}
|   ano not in {2011}
|   |   mes in {11,12,1,2,3}: 2.181 {}

```