Memoria

Trabajo Fin de Máster

BIGDATA

SMARTPRICING

Autor: Teresa Alaguero Garcia

Agradecimientos.

Quisiera aprovechar para agradecer a todas y cada una de las personas que nos han ayudado con sus enseñanzas, ofreciéndonos una nueva visión de nuestras propias capacidades.

Por todos los momentos perdidos con las personas que realmente importan, por su capacidad de comprensión y sacrificio, manteniendo sus necesidades a un lado mientras me dedicaba por completo a este proyecto.

Por no dejar que perdiésemos la visión del objetivo a conseguir.

Resumen

Hoy en día la satisfacción del usuario y la conectividad para con las compañías que ofrecemos servicios se antoja un punto muy importante para la supervivencia de toda relación saludable. El beneficio de que las necesidades tanto de los clientes como de la propia compañía estén en consonancia es, sin duda, el objetivo a perseguir por cualquier empresa.

Para alinear dichas necesidades se propone la creación de una nueva relación cliente/compañía ofreciendo a nuestros usuarios una nueva plataforma para conseguir la simulación de un seguro en un tiempo menor al que actualmente deberíamos emplear en este tipo de trámite y sobre todo sabiendo que mediante las tecnologías de las que hoy disponemos y la capacidad del análisis de los datos, ofreceremos siempre un mejor precio, así como un mejor servicio a todos los clientes que se decidan a probar esta nueva propuesta.

Esperamos que la consecución de estos dos objetivos nos coloque en una nueva posición, tanto en la satisfacción del cliente como en el desarrollo del negocio en el que REALE seguros pretende ser líder.

En este documento se explica el por qué de la realización de este proyecto después de haber estudiado las necesidades tanto de los clientes como de la empresa y proponer las soluciones que se detallan en los siguientes apartados.

Contenido

Índice de Figuras.

Índice de Tablas.

Índice de Gráficos.

Índice de Anexos.

1. Introducción

Actualmente y tras haber realizado un estudio sobre las aplicaciones que existen hoy en el mercado para la simulación de un seguro, se observa que muchas de aquellas búsquedas que se suponen inicialmente “sencillas”, pueden llegar a ser tediosas y, en muchos casos, al solicitar una cantidad importante de datos personales, hace que los clientes abandonen dicha tarea por aburrimiento o por un simple motivo de privacidad. Esta situación, supone en los inicialmente “potenciales clientes”, un incremento en el índice de abandono.

Lo que se pretende con este proyecto es crear “clientes fidelizados”, ofreciendo un servicio ágil e intuitivo para cualquier persona, ya que hoy en día las nuevas tecnologías, en muchas ocasiones, no parecen estar hechas para todos.

El alcance del proyecto se centró en obtener un software escalable y de uso apto para cualquier persona, y que además de su facilidad de uso, sea capaz de ofrecer al cliente el precio óptimo y maximizando el margen por cliente para Reale.

A lo largo de los capítulos de esta memoria, se irán describiendo las fases de la que consta el proyecto y el procedimiento que se utiliza para superar cada uno de los objetivos que se propone mejorar.

2. Objetivos

Los objetivos de este proyecto han quedado claramente marcados en dos:

1.- La creación de una plataforma con una interfaz más sencilla que permita a todo tipo de clientes utilizarla. De esta forma mejoramos la experiencia de usuario, que hoy en día puede proporcionar un antes y un después en la relación con nuestro principal activo.

2.- La optimización del precio ofrecido, que genere el precio final del seguro óptimo para el cliente y con el mejor margen para Reale. Para encontrar este punto de equilibrio en todas las cotizaciones se llevan a cabo dos procesos, uno de clasificación y otro de optimización. El proceso de clasificación, que se realiza sobre los resultados de los algoritmos de regresión, consiste en definir un score que cumple el papel de clasificador en función del margen (algoritmo regresivo que predice el posbile margen que nos dejaría un cliente con determinadas carácterísiticas) que da la compañía. Y cada vez que se hace una cotización, se realiza el proceso de optimización, que consiste en evaluar la diferencia entre la prima actualmente en los sistemas, calculadas por el método tradicional (GLM) por Reale y la ofrecida por la competencia, teniendo en cuenta la probabilidad de conversión, con el fin de ajustar el precio final que consigue maximizar el margen manteniendo la competitividad.

Con estos dos objetivos se proporciona agilidad a la hora de realizar una simulación, y ofrece una mejor experiencia de usuario.

3. Descripción Del Proyecto

El proyecto de Smart Pricing que consiste en generar una prima optimizada, se aborda con tecnología Big Data para ofrecer una solución robusta y escalable que garantice que los procesos desarrollados para el cumplimiento de los objetivos propuestos anteriormente se ejecuten con un alto performance.

3.1 Definición de KPIs

Los **KPI** o indicadores de desempeño clave son métricas (medibles y cuantificables) que determinarán numéricamente una variable (por ejemplo: ingresos, gastos, número de consultas…) directamente relacionada con los objetivos marcados dentro de la estrategia o plan actual. Son un valor tangible que permite medir y cuantificar si se está en el camino correcto para lograr un objetivo.

Existen diversos tipos de KPI pero tendremos que marcar cuales son necesarios para el negocio. Deben cumplir los siguientes requisitos:

- Ser medible
- Cuantificable
- Debe ser periódico o temporal
- Específico
- Relevante

Indicadores clave de desempeño (KPIs)

Los KPIs que se determinan en este proyecto, desde una perspectiva de negocio aparecen reflejados en la tabla 1. Indicadores clave de desempeño. Calculamos un tiempo de seis meses para comprobar la evolución de dichos indicadores en el negocio.

Tabla 1, Indicadores clave de desempeño (KPIs)

Indicador	Descripción
Nº Clientes nuevos (no eran de Reale)	Clientes nuevos que no pertenecían a Reale
Nº Clientes fidelizados 1.	Clientes antiguos de Reale, que permanecen sin estar en la tabla de Scoring
Nº Clientes fidelizados 2	Clientes antiguos de Reale, que permanecen por el scoring aplicado favorable
Nº Clientes abandonan Reale	Clientes que abandonan Reale
Nº Usuarios utilizan la aplicación	Clientes y no Clientes
Nº Usuarios por tramos de edad	Usuarios de la aplicación por tramos de edad: >=18 y <25 ; >=25 y <40 ; >=40 y < 55 ; <=55

Nº Clientes por zona demográfica	Según código postal.
Porcentaje de aplicación por cada factor existente	Para los 9 factores que se aplican
Nº solicitudes mejoradas por Scoring	Cuando el Scoring mejora el precio.
Nº Solicitudes no mejoradas por Scoring	Cuando el Scoring incrementa el precio.

3.2 Metodología

Para poder extraer conocimiento a partir de grandes volúmenes de datos, se han creado, a lo largo de estos últimos años distintos modelos de proceso para el descubrimiento de conocimiento y minería de datos, llamados KDDM process model (Knowledge Discovery and Data Mining process models).

Uno de los modelos de referencia que más apoyo ha tenido de las empresas privadas y organismos públicos, es CRISP-DM (Cross Industry Standard Process for Data Mining), como puede observarse en la siguiente gráfica (figura 3). Y aunque ha experimentado un ligero descenso en los últimos años, sigue siendo la más empleada de las distintas metodologías.

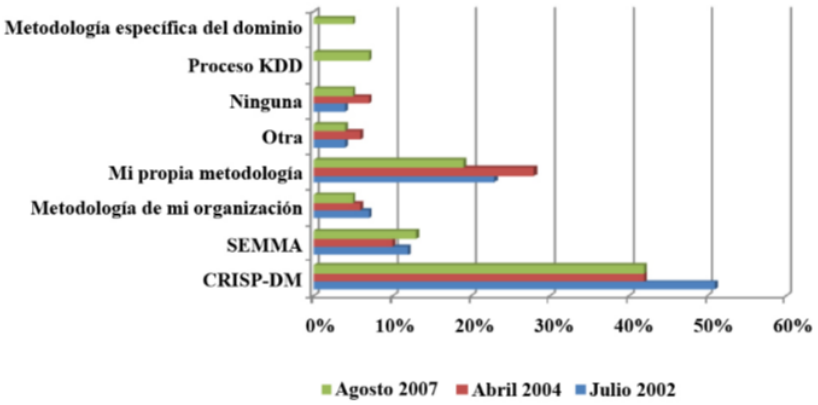


Figura 1. Utilización modelo CRISP-DM.

CRISP-DM está compuesto por 6 etapas, que son entendimiento del negocio, entendimiento de los datos, preparación de los datos, modelado, evaluación y despliegue. Estas etapas a su vez tienen distintas subfases. En la figura 3 se pueden observar las distintas fases y las posibles secuencias.

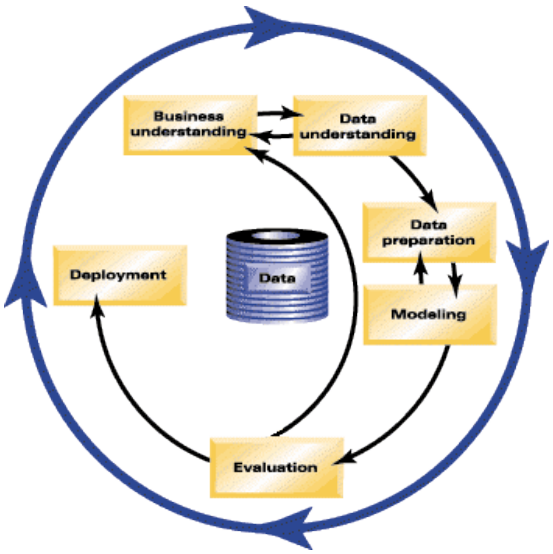


Figura 2. Modelo CRISP-DM, fases.

3.3 Planificación y Cronograma

Finalmente, esta tarea de la primera fase de CRISP-DM, tiene como meta desarrollar un plan para el proyecto, que describa los pasos a seguir.

1. Entendimiento del Negocio y Gestión del desarrollo del Proyecto	2. Entendimiento de los datos	3. Preparación de los datos	4. Modelado y Desarrollo del proyecto	5. Evaluación	6. Despliegue
1.1 Reunión con el personal interno de Reale para especificar objetivos del proyecto 1.2 Desarrollar planificación dell Proyecto 1.3 Reuniones de seguimiento y Control de cambio 1.4 Gestionar equipo del proyecto	2.1 Recolección de los datos 2.2 Descripción de los datos 2.3 Exploración de los datos 2.4 Verificación calidad de los datos	3.1 Estructuración de los datos 3.2 Integración de los datos 3.3 Formateo de los datos 3.4 Eliminación datos erróneos 3.5 Creación y agrupación de variables	3.1 Creación App y Web 3.2 Desarrollo software 3.3 Pruebas software 3.4 Desarrollo algoritmos 3.5 Pruebas algoritmos	5.1 Evaluación de Plataformas 5.2 Incluir de la memoria parte de evaluación	6.1 Generación entorno Azure 6.2 Generación App 6.3 Generación Web 6.4 Generación y Despliegue de la API 6.5 Pruebas algoritmos
Investigación y Documentación					

Tabla 2. Plan de Proyecto.

Por otro lado, también se elabora un cronograma con las tareas definidas en el Plan de Proyecto.

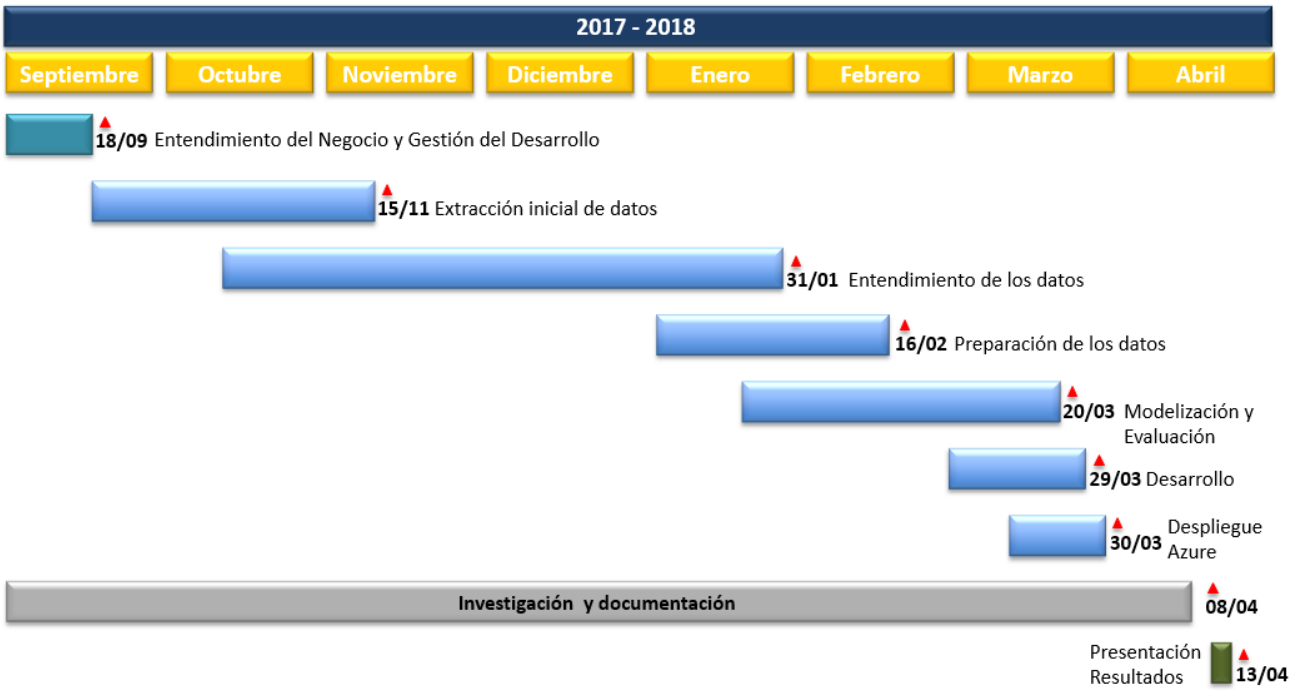


Tabla 3. Cronograma Planificación del Proyecto

4. Tecnología Cloud

El Cloud es un modelo para proveer servicios de infraestructura, plataforma y aplicaciones bajo demanda. Este modelo provee agilidad, escalabilidad, tolerancia a fallos y eficiencia; todos estos elementos son los que buscamos incorporar en el desarrollo de este proyecto.

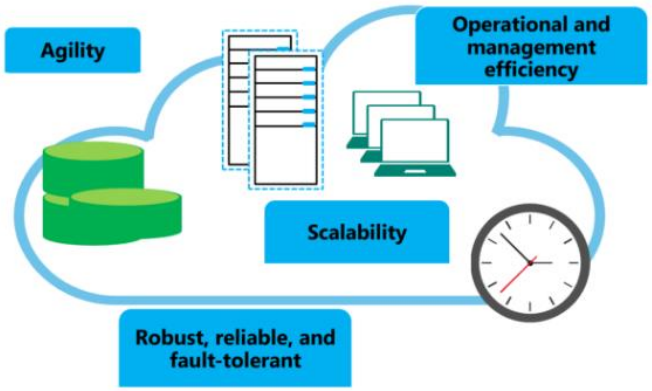


Figura 3. Tecnología Cloud

4.1 Soluciones en nube

Las oportunidades de negocio y de mercado pueden aparecer y cambiar muy rápidamente una oportunidad de negocio hoy puede no ser relevante mañana. La plataforma informática debe responder de forma ágil y rápida a estos cambios de negocio facilitando la competitividad. Las soluciones en la nube permiten hacer todo esto.

La economía también puede ser un factor importante para la selección de la nube. En muchos casos es mucho más económico utilizar un servicio en nube que construir controlar y mantener servicios por sí mismo. La nube permite un uso más eficiente de los recursos lo cual reduce los costes en general.

4.1.1 Soluciones en nube características esenciales

Independientemente de las tecnologías específicas que las organizaciones utilizan para implementar computación en la nube el Instituto Nacional de estándares y tecnología de Estados Unidos (National Institute of Standards and Technology (NIST)) ha identificado cinco características esenciales que son parte de una solución en la nube:

Servicios en demanda. La habilidad que tiene el usuario de generar nuevos recursos por sí mismo sin la ayuda del proveedor de servicios.

- **Acceso de banda ancha**. Que los recursos sean accesibles por medio de la red pública sin necesidad de infraestructuras especiales

- **Reserva de recursos**. Que existen recursos en reserva para ser utilizados en momentos puntuales y volver a su estado inicial en cualquier momento.
- **Rápida elasticidad**. La habilidad para aumentar o disminuir los recursos de manera manual o automática sin que esto involucre tiempos de espera.
- **Servicio medido**. El poder medir exactamente qué recursos están siendo consumidos, monitorizarlos y controlarlos.

4.2 Módulos de la Nube

Se pueden configurar en diferentes servicios a través de diferentes modelos de implementación. Existen nubes públicas nubes privadas y una mezcla de las dos usando un modelo híbrido.

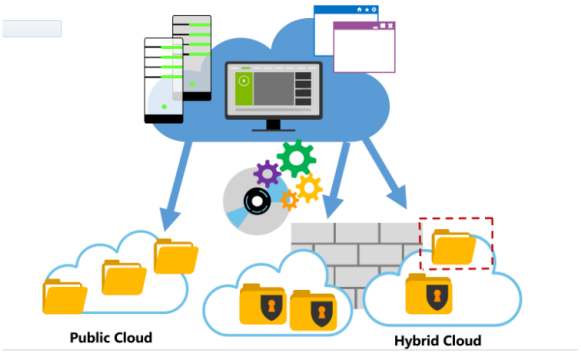


Figura 4. Módulos de la Nube

4.2.1 Nube Pública

Una nube pública es propiedad de un proveedor de servicios. Este proveedor entrega recursos a una organización que es el cliente final El cual se conecta por una conexión segura de red típicamente sobre internet. El proveedor de servicio comparte sus recursos con múltiples organizaciones o con el público en general. La principal característica es que todos los recursos pertenecen a este proveedor.

4.2.2 Nube Privada

Una nube privada opera sólo dentro de una organización en una red privada y es altamente segura. Provee funcionalidades de nube a departamentos internos específicos. Una nube privada es un conjunto de recursos en reserva o en uso que los diferentes clientes usan y comparten.

4.2.3 Nube Híbrida

La nube híbrida es una combinación del modelo público y el vuelo privado. en una nube híbrida los recursos específicos pueden utilizarse en la nube pública y otros en la nube privada. La nube híbrida se beneficia de la seguridad de la nube privada y de la flexibilidad de la nube pública.

4.3 Tipos de servicio en la Nube

Hay diferentes modelos para los servicios en la nube dependiendo en como en servicio es utilizado o proveído. En términos de los tipos de servicio estos pueden dividirse en tres categorías principales:

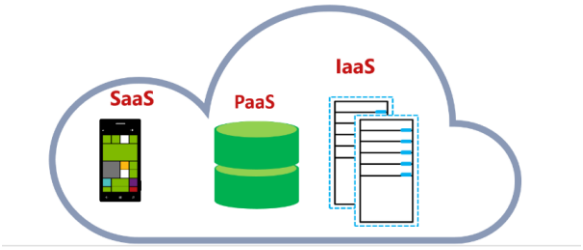


Figura 5. Tipos de servicio en la Nube

4.3.1 Software como Servicio (SaaS)

SaaS son aplicaciones de software que se proveen como servicio de la nube. el usuario se suscribe al servicio y lo utiliza a través de un navegador o instalando una aplicación cliente.

Ejemplos de SaaS son por ejemplo incluye Office 365 o Skype. La principal ventaja de esto es que permite al usuario acceder fácilmente a la aplicación sin necesidad de instalarla y mantenerla. Típicamente los usuarios no tendrían por qué preocuparse de actualizaciones o mantenimientos ya que el proveedor se encarga de todo ello

4.3.2 Infraestructura como Servicio (IaaS)

La infraestructura como servicio es cuando el proveedor es capaz de dar servicios virtualizados como servidores, red, almacenamiento y componentes que fácilmente se habilitan o deshabilitan a conveniencia.

Un punto importante es que un servicio de infraestructura puede ser incluso un único servicio por ejemplo un servidor virtual que tiene una instalación de una base de datos. También puede ser un conjunto de infraestructura pre-configurada para un ambiente predeterminado. La organización puede definir un grupo de máquinas virtuales y plantillas de red que pueden provisionarse como si fuera una unidad única que hace conjunto completo.

4.3.3 Plataforma como Servicio (PaaS)

Se habla de plataforma como servicio cuando el proveedor es capaz de provisionar recursos para que los desarrolladores construyan sus propias soluciones. Típicamente proveer capacidad de sistema operativo almacenamiento y computación para desarrollar aplicaciones móviles, fijas, interfaces web. El proveedor provee APIs (Application Programming Interface) que simplifica la creación de soluciones.

4.4 Máquinas Virtuales

Existen muchas opciones para ejecutar aplicaciones en la nube. Una de ellas es el uso de máquinas virtuales.

Una máquina virtual es un servidor funcionando en la nube que hace uso de un grupo de servicios como almacenamiento, redes virtuales y directorios.

Una máquina virtual provee la flexibilidad de la virtualización sin requerir todos los gastos de comprar y mantener su propio centro de datos.

Figura 6. Máquinas Virtuales en la Nube

4.5 Servicios de Azure en la nube

Un servicio de infraestructura en la nube es el contenedor de red que aloja las máquinas virtuales. Cualquier máquina virtual en un servicio en la nube puede comunicarse directamente con cualquiera de las otras máquinas virtuales en ese servicio utilizando las comunicaciones internas.

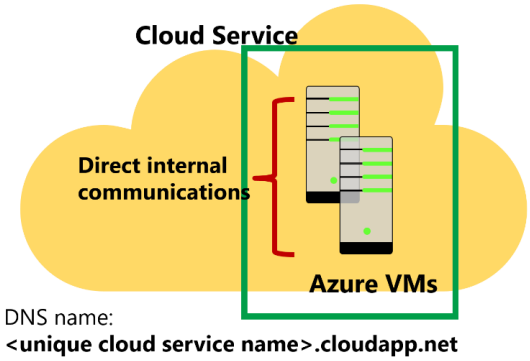


Figura 7. Servicios de Azure en la Nube

4.5.1 Servicio Azure App

El servicio de Azure app o Azure App Service es un servicio integrado que le permite crear aplicaciones web y móviles para cualquier plataforma o dispositivo que se integra completamente con las demás soluciones de la nube como Office 365 o Dynamics CRM**.**

Figura 8. Servicio Azure App

4.5.2 Uses for Microsoft Azure

Microsoft Azure ofrece muchos servicios y recursos. Por ejemplo, se pueden usar las máquinas virtuales de Azure para construir una red de servidores virtuales que alojen una aplicación, una base de datos o una solución hecha a medida que puede ser una oferta de infraestructura como servicio. Otros servicios pueden ser categorizados como plataforma como servicio ya que se usan sin hacer mantenimiento de los sistemas operativos

Por ejemplo, cuando usted ejecuta un sitio web en Azure Web Apps o una base de datos SQL en Azure SQL no es necesario asegurarse que se está utilizando la última versión de Internet Information Services o de SQL Server y que están los últimos parches y actualizaciones, es responsabilidad de la plataforma

4.5.3 Servicios Azure: Computación, Almacenamiento e Identidad

Microsoft Azure provee servicios en la nube para llevar a cabo varias tareas y funciones a través de todo el espectro de la tecnología de la información. Estos servicios se pueden organizar en varias categorías:

Figura 9. Servicios Azure: Computación, Almacenamiento e Identidad

4.5.3.1 Servicios de Computación y de Red

- **Virtual Machines.** Máquinas virtuales Windows o Linux que se crean desde plantillas predefinidas o que se implementan desde imágenes creadas a medida.
- **Virtual Machine Scale Sets.** Son conjuntos de máquinas virtuales balanceadas que pueden ser activadas de forma simultánea.
- **Virtual Networks**. Redes virtuales para conectar las máquinas virtuales.
- **Cloud Services**. Servicios en la nube de plataforma como servicio multinivel que se pueden implementar y manejar en Microsoft Azure.
- **Load Balancer**. balanceadores de carga que fácilmente y rápidamente permiten escala las aplicaciones dando soporte a la mayoría de los protocolos de red más comunes.
- **VPN Gateway**. Herramienta para conectar redes Azure a través de VPN usando protocolos seguros como IPSec o IKE.
- **Azure DNS**. Servicios de nombre de dominio para poder alojar sus servidores en Azure.
- **ExpressRoute**. por medio de este servicio se pueden crear conexiones dedicadas de alta velocidad entre servidores en el sitio el cliente y Azure.
- **Traffic Manager**. implementación de sistemas de balanceo de carga para dar alta escalabilidad y disponibilidad.
- **Network Watcher**. herramientas para monitorizar y diagnosticar problemas de la red.

4.5.3.2 Servicios de almacenamiento y copias de seguridad

- **Azure Storage**. Servicio para almacenar información en ficheros, objetos, tablas y colas.
- **Data Lake Store**. Es el repositorio de gran tamaño que permite almacenar las cargas de trabajo para el análisis de Big Data.
- **StorSimple**. Infraestructura de almacenamiento consolidado que automatiza la administración de la información y acelera la recuperación en caso de desastre.
- **Backup.** Azure puede ser utilizado como el destino por defecto para almacenar copias de seguridad de servidores que estén en las instalaciones del cliente.
- **Azure Site Recovery**. Administración completa de fallos tanto en cliente como en estructuras privadas de nube privada de Azure.

4.5.3.3 Servicios de seguridad e identidad

- **Security Center**. Con esta herramienta se puede tener una vista centralizada del estado de toda la seguridad de los recursos que se están utilizando en Azure.
- **Key Vault**. Por medio a este servicio se pueden crear e importar llaves de encriptación, reducir la latencia y simplificar las tareas de los certificados SSL/TLS certificates.
- **Azure Active Directory**. Integración del Directorio corporativo con los servicios de la red que facilitan el single sign on (SSO).
- **Azure Multi-Factor Authentication**. Herramienta para implementar medidas de seguridad adicionales en las aplicaciones con el fin de verificar la identidad del usuario.

4.5.4 Servicios Azure: Web, información, multimedia y gestión

Figura 10. Servicios Azure Web, Información, multimedia y gestión

4.5.4.1 Servicios web y móviles

- **App Service**. Por medio este servicio se pueden crear aplicaciones escalables en nube para Web y para móvil sin necesidad de manejar o administrar o gestionar la configuración de servidores web.
- **Web Apps**. Facilita la creación e implementación de webs de misión crítica.
- **Mobile Apps**. Implementa el servicio de back-end para aplicaciones móviles que se pueden ejecutar en múltiples plataformas.
- **API Apps**. Por medio de esta herramienta se pueden publicar sus APIs de forma segura.
- **Logic Apps**. Herramienta que automatiza el acceso y el uso los datos a través de las nubes sin tener que escribir código.
- **Content Delivery Network**. Servicio para asegurar la entrega de contenido de forma segura y fiable con alcance global.
- **Media Services**. Codifique almacene y distribuya vídeo o audio a cualquier escala.
- **Azure Search**. Herramienta de búsqueda.

4.5.4.2 Servicios de base de datos información y analítica

- **SQL Database**. Herramienta para implementar bases de datos relacionales para las aplicaciones sin la necesidad de aprovisionar y administrar un servidor de base de datos.
- **SQL Data Warehouse**. Servicio de bases de datos relacionales con la capacidad de procesamiento paralelo.
- **Azure Cosmos DB**. Servicio para implementar una base de datos en Azure Cosmos DB que funciona como base de datos distribuida global usando APIs multimodelo.
- **HDInsight**. Use Apache Hadoop para llevar a cabo procesamiento y análisis de Big Data.
- **Redis Cache**. Caché de alto desempeño para las aplicaciones.
- **Machine Learning**. Aplique modelos estadísticos a sus datos y lleve a cabo análisis predictivo desempeño.

4.5.4.3 Servicios de monitorización y gestión

- **Microsoft Azure Portal**. Construya Administrator todos los productos de Azure desde una única consola.
- **Azure Resource Manager**. Use Azure Resource Manager para desplegar administrar y monitorizar los componentes de infraestructura y los recursos necesarios para las aplicaciones y los servicios.
- **Log Analytics**. Centralice los registros de múltiples sistemas en un único almacenamiento ganando mucha mayor visibilidad de todo su entorno.

- **Automation**. Herramienta para simplificar la administración de la nube con automatización de procesos.
- **Scheduler**. Programa y monitoriza tareas y acciones recurrentes.

5. Plataformas BIG DATA

En este apartado comentaremos las plataformas Big Data que se han probado en el desarrollo de este proyecto.

- Plataforma Oracle Big Data Cloud Service
- Plataforma On-premise
- Plataforma Azure

A continuación, describimos los componentes de hardware y software con los que contábamos en cada una de ellas:

5.1 Plataforma Oracle Big Data Cloud Service

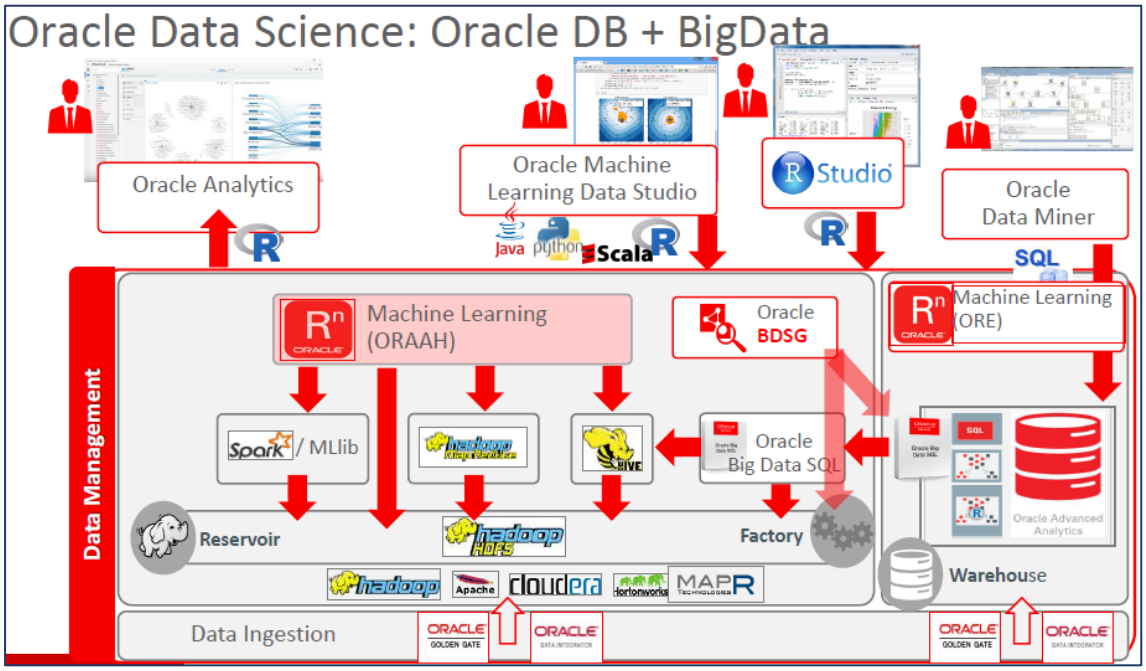
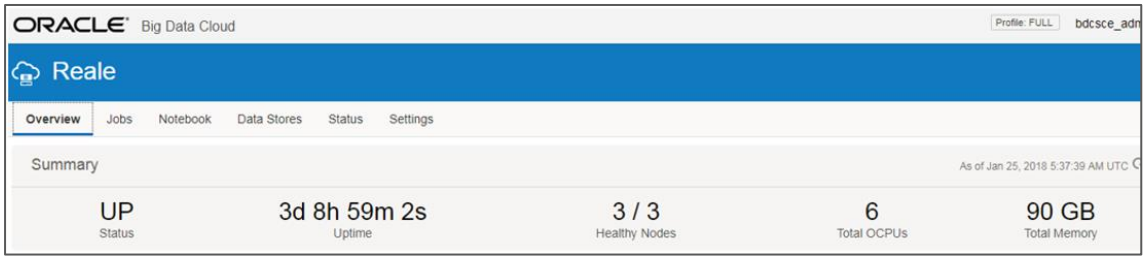


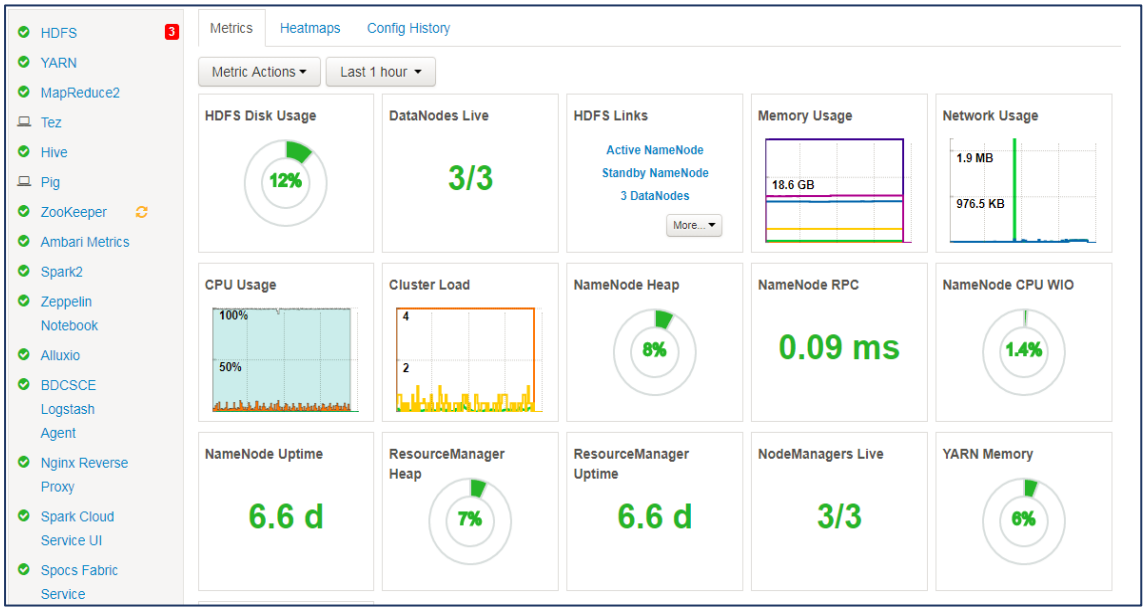
Figura 11 Plataforma Oracle BigData Cloud Service

SmartPricing

Para realizar pruebas contamos con acceso a Oracle Big Data Cloud Service -Compute Edition, durante dos meses, con la siguiente configuración:

- Distribución Horton de Hadoop, con un clouster de 3 Nodos, 6 Cpus y 90GB de RAM
- IDE utilizado RStudio
- Notebook utilizado Zepelin
- Interface Analítica de cluoud computing H2o ia





Este servicio lo probamos durante 2 meses sin coste alguno. Oracle ofrece este servicio en modalidad de contratación mensual o anual, sin condiciones de permanencia alguna y los importes para la configuración probada se describen en la siguiente tabla:

Oracle Big Data Cloud Services							
Oracle Big Data Cloud Service - Compute Edition	**Pay as You Go**	**Price**	**Metric**	**Metric Minimum**	**Units**	**Cost Moth**	**Yearly Cost**
Oracle Big Data Cloud Service - Compute Edition - Compute Capacity	0,2101	0,1400	OCPU Per Hour	2	16,00	1.666,56	19.998,72
Oracle Big Data Cloud Service - Compute Edition - Storage Capacity	0,0613	0,0409	Gigabyte Storage Capacity per Month	-	1.000,00	40,90	490,80
Oracle Cloud Infrastructure - Object Storage Classic							
				TOTAL		1.707,46	20.507,52

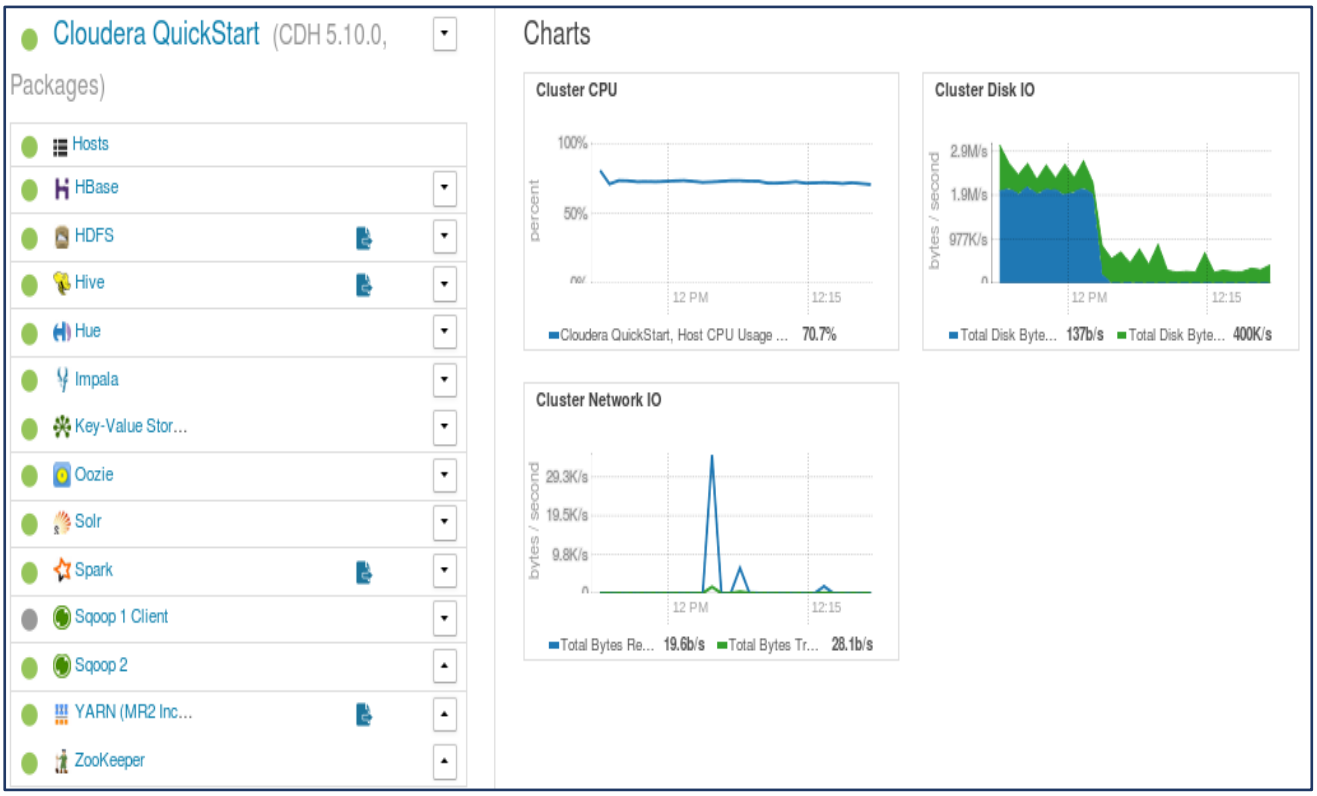
Tabla 4. Costes plataforma Oracle.

5.2 Plataforma On-Premise

En modo laboratorio de manera temporal contamos con una máquina virtual configurada con 32Cores, 256 GB RAM y 1TB Almacenamiento.

Software Instalado:

- Sistema Operativo: Centos 7.0
- Hadoop Distribución de Cloudera Quick Start DHC 5.10.0
- IDE utilizado RStudio.
- Interface Analítica de cloud computing H2o ia



Consumo de CPU y Memoria en tiempo de entrenamiento de algoritmos

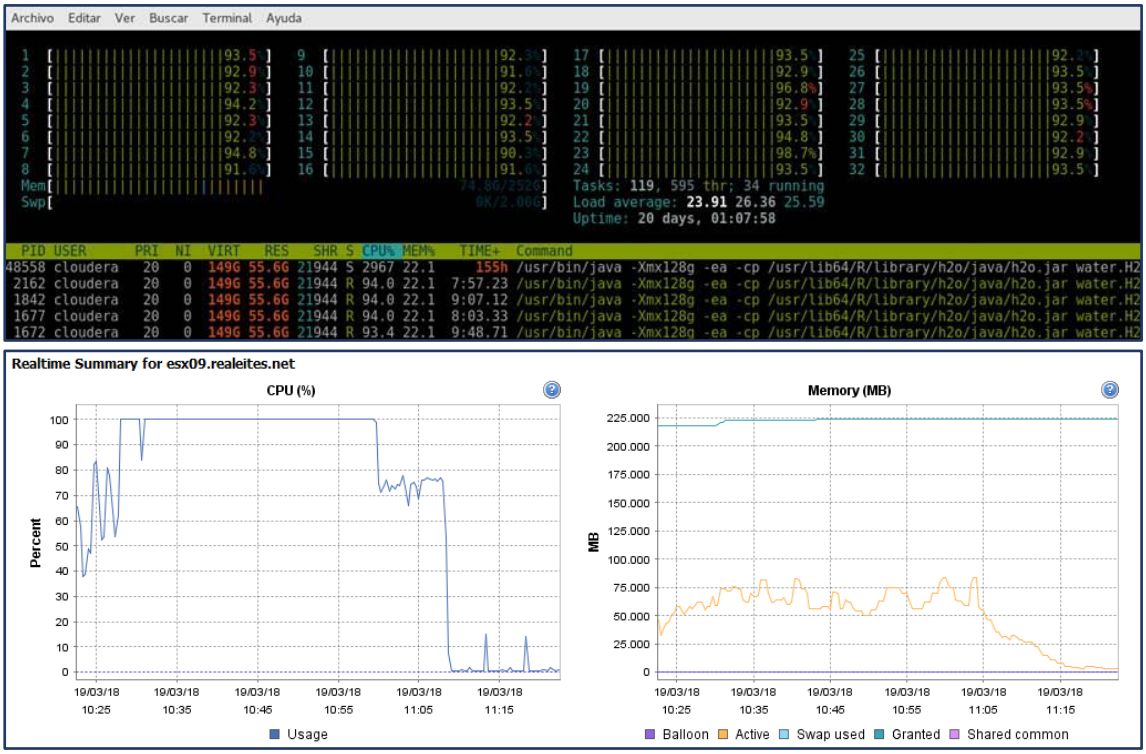


Figura 12. Plataforma On-Premise

Por motivos de pérdida de la primera máquina desplegada, ajenos a la operativa normal del entorno de pruebas, durante el último mes se ha contado con una segunda máquina, para repetir el entrenamiento de los modelos en simultáneo.

Esta segunda máquina está configurada con 16Cores y128GB de RAM.

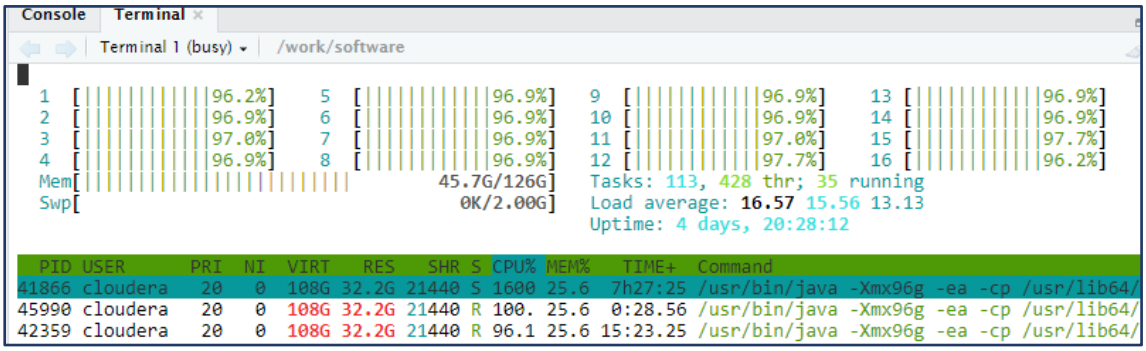


Figura 13. Segunda mqáuina On_Premise

En las siguientes gráficas podemos observar el funcionamiento del host con las dos máquinas ejecutando algoritmos:

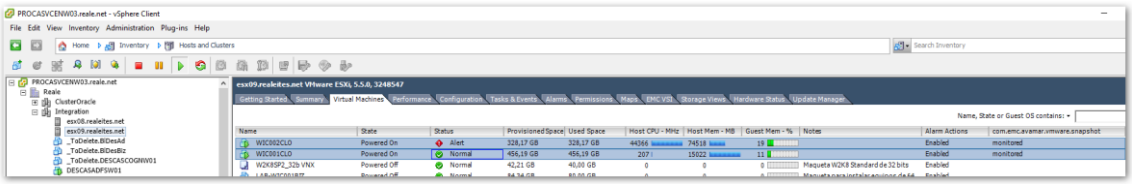


Figura 14. Funcionamiento del host con dos máquinas On-Premise

Servidor 1:

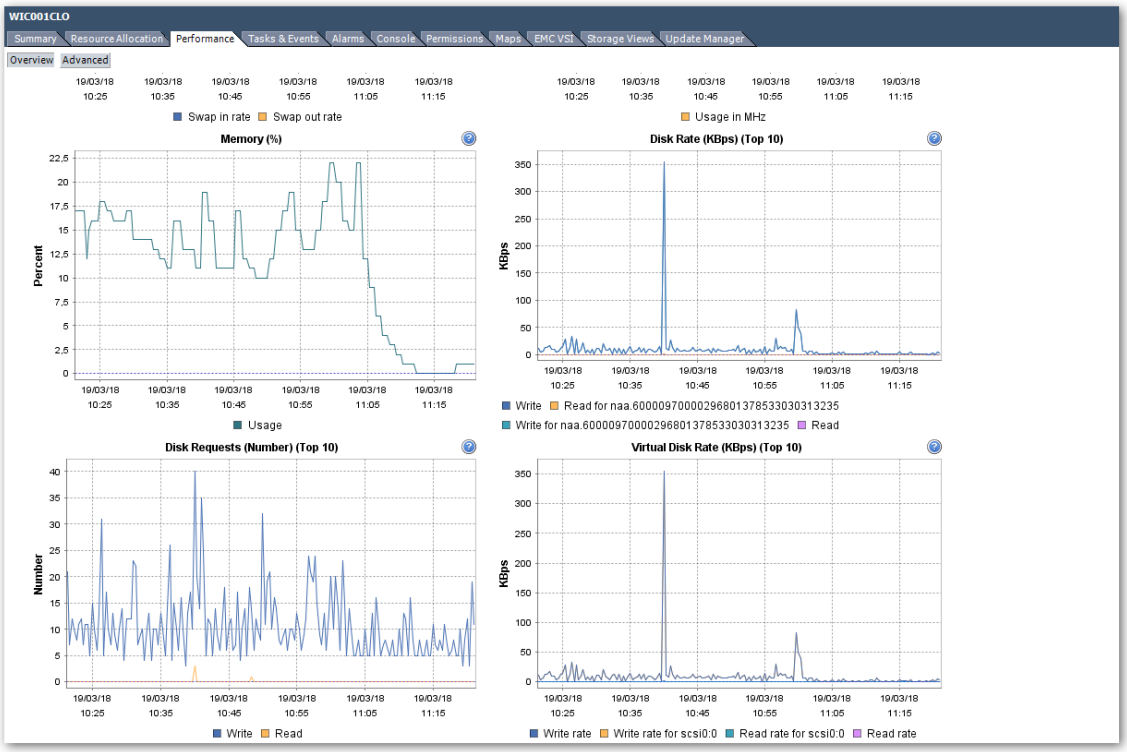


Figura 15 Vista 1 Servidor 1-Plataforma On-Premise

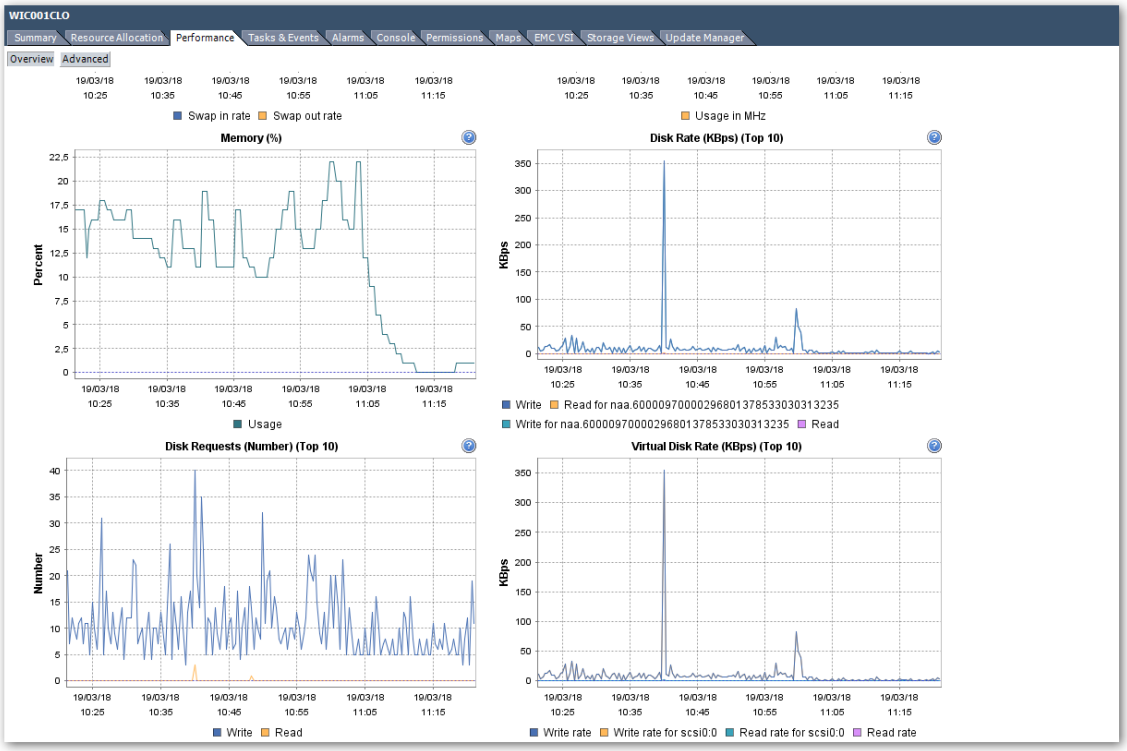


Figura 16 Vista 2 Servidor 1-Plataforma On-Premise

Servidor 2:

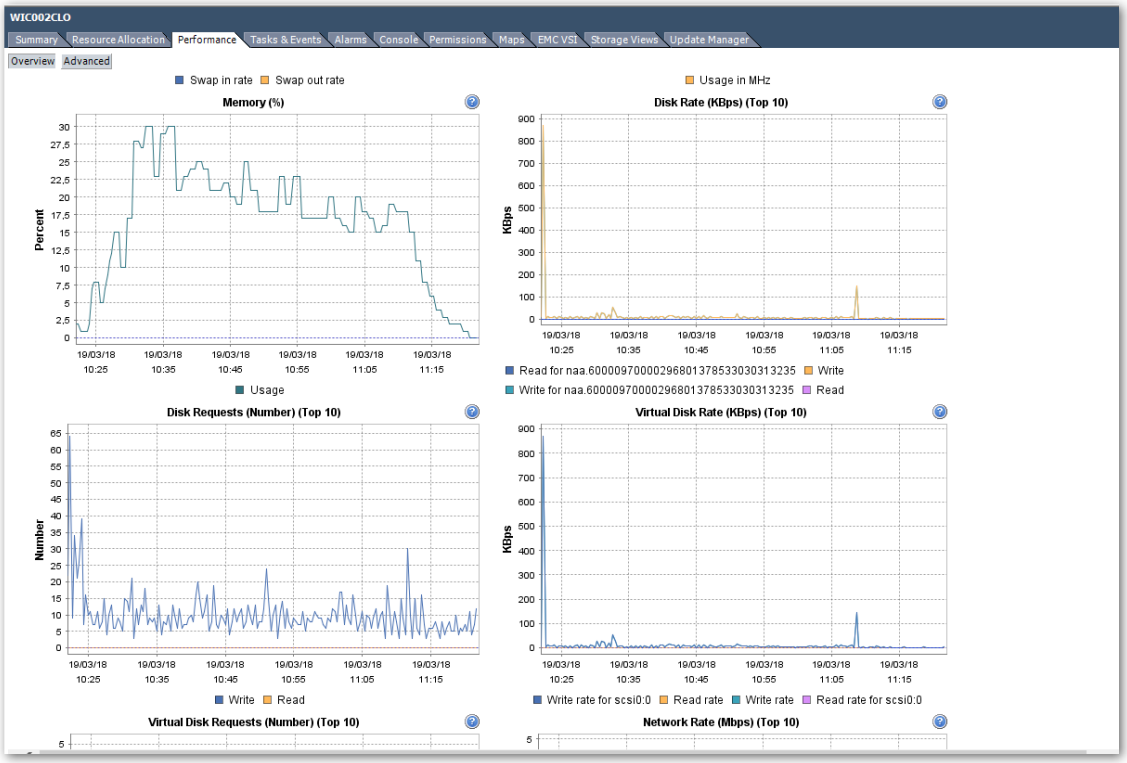


Figura 17 Vista 1 Servidor 2-Plataforma On-Premise

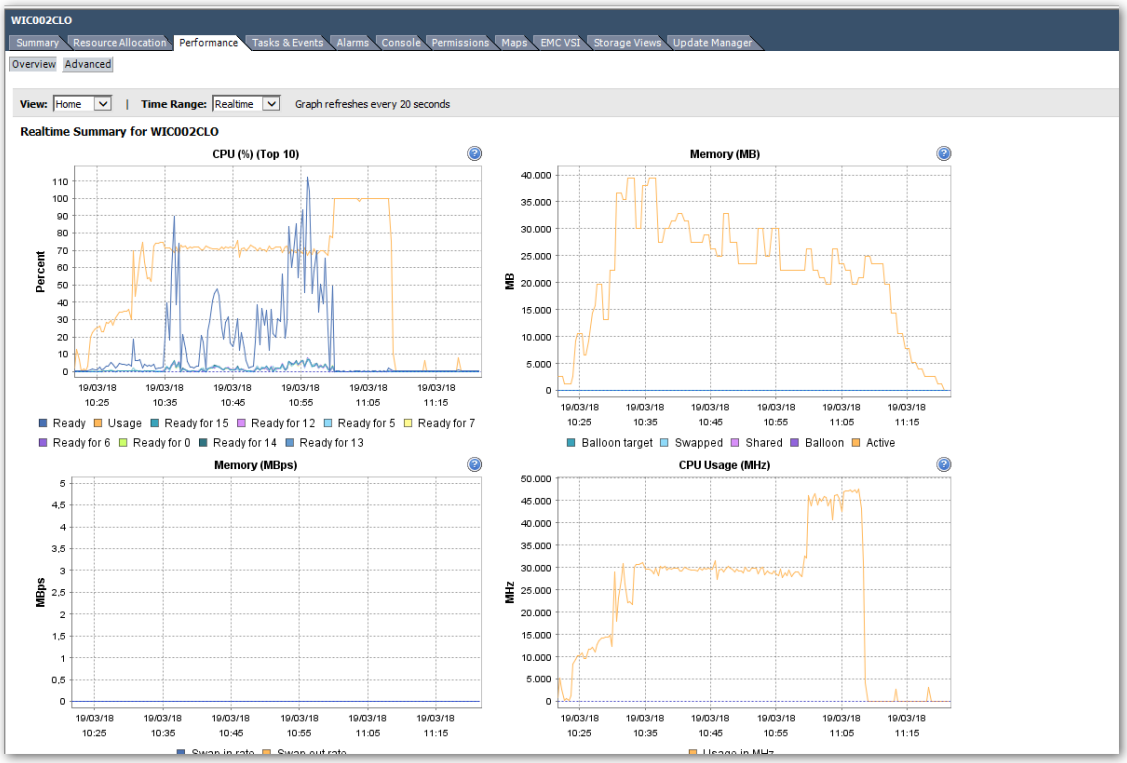


Figura 18 Vista 2 Servidor 2-Plataforma On-Premise

Los costes de adquisición de infraestructura se pueden ver en la siguiente tabla:

Recurso	Coste
RAM 256 GB	8.320 €
CPU 32 Cores	6.350 €
RAM128 GB	4.220 €
CPU 32 Cores	3.150 €
Almacenamiento 4TB	4.000 €
Total Coste Adquisición	**26.040 €**

Tabla 5. Costes Infraestructura On-Premise

5.3 Plataforma AZURE

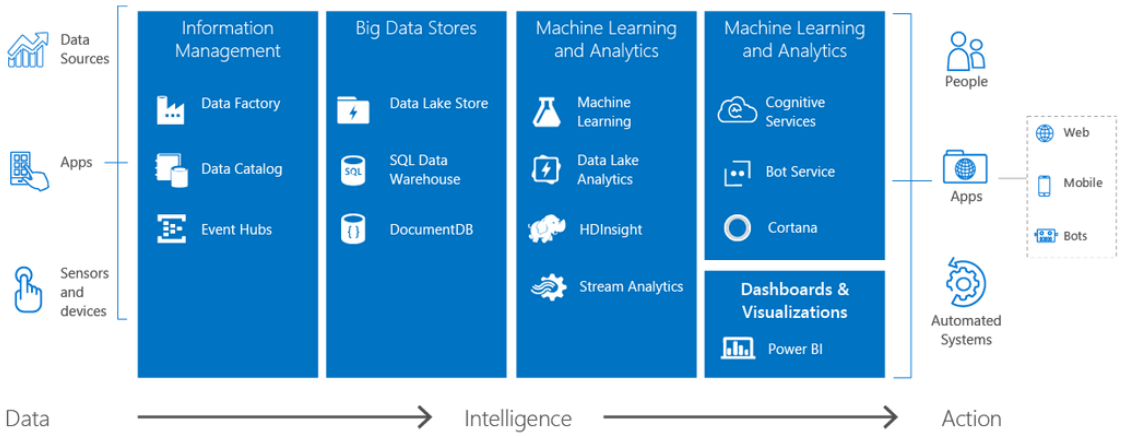


Figura 19. Vista Plataforma Azure

En Azure se cuenta con una completa gama de recursos, que se pueden desplegar de una manera muy sencilla e inmediata, en la siguiente gráfica se puede observar algunos de los recursos probados en el desarrollo de este proyecto

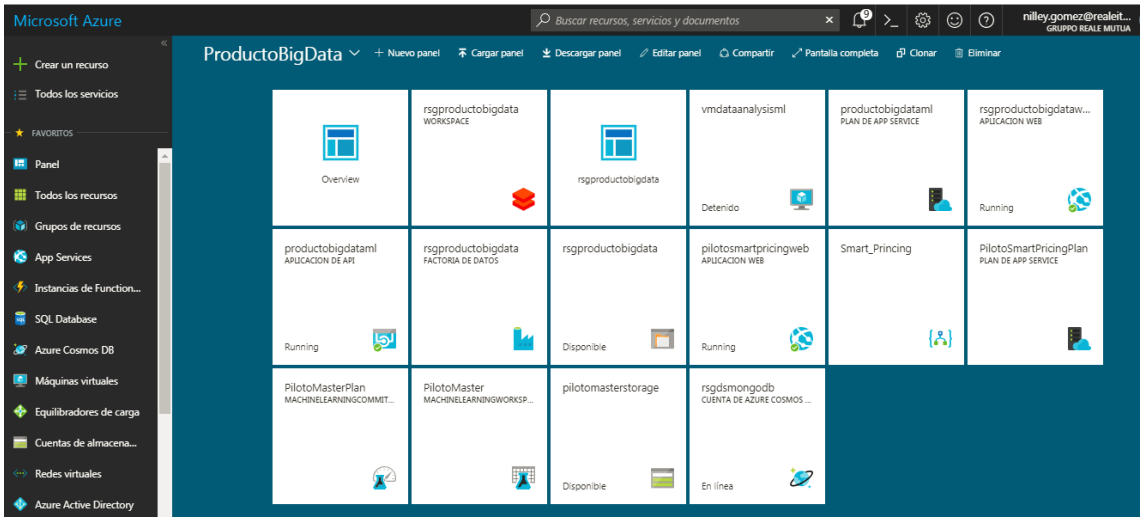


Figura 20. Recursos probados en Plataforma Azure

Utilizamos Azure Cosmos DB, para almacenar el resultado de cada uno de los Webservices que recopilaba información requerida para los algoritmos de Machine Learning y para los procesos de clasificación y optimización finales.

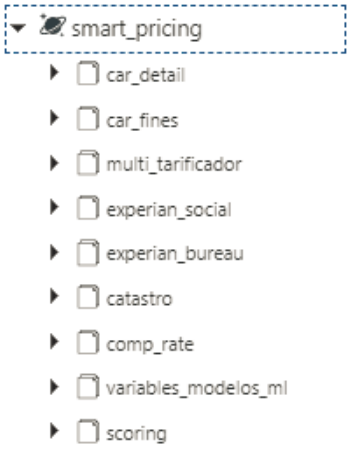


Figura 21. Colecciones Azure CosmosDB

Se optó por esta base de datos, con la finalidad de probar el uso de, bases de datos no tradicionales, un propósito exclusivamente de investigación, así como el conocer los costes de tarificación generados.

Total cost
147.33 EUR

Search to filter items...

NAME	TYPE	RESOURCE GROUP	COST (EUR)
rsgproductobigdata	Data factory (V2)	ProductoBigDataRG	0.04
rsgproductobigdata	Data Lake Store	ProductoBigDataRG	0.00
rsgdsmongodb	Azure Cosmos DB account	ProductoBigDataRG	129.51
rsgproductobigdata	Storage account	ProductoBigDataRG	0.01
productobigdataml	App Service plan	ProductoBigDataRG	17.74
ServicePlan38318aba-aa4e	App Service plan	ProductoBigDataRG	0.03
productobigdataml	App Service	ProductoBigDataRG	0.00

Tabla 6. Consumos BBDD en Azure

La anterior tabla se corresponde al consumo de un mes de los recursos desplegados y como se puede observar, el consumo más alto se corresponde a Cosmos DB, en el que tan solo se tienen 9 colecciones. Por este motivo, una vez finalizada esta prueba Piloto, se recomendará cambiar a almacenamiento en bases de datos tradicionales como por ejemplo SQL Server u Oracle.

En la siguiente tabla, detallamos los recursos desplegados y los costes asociados a cada uno:

Recurso	Proposito	Coste Mes	Tiempo	Coste Total
VM A7 (8Cores 56GB)	Entrenamiento Modelos	1.011 €	20 Días	674 €
VM L32 (32Cores 256GB)	Entrenamiento Modelos	2.214 €	40 Horas	119 €
APP Service B1 (1Core, 1.75GB RAM)	Web	55 €	4 Meses	220 €
APP Service B1 (1Core, 1.75GB RAM)	API Modelos ML	55 €	2 Meses	110 €
Storage General Purpose V1	almacenar 1 GB	24 €	3 Meses	72 €
Cosmos DB	9 Colecciones	207 €	4 Meses	832 €
Area Trabajo Machine Learning Studio		8 €	5 Meses	40 €
TOTAL				**2.067 €**

Tabla 7.Costes recursos desplegados en Azure

Por último, realizamos una comparativa de los costes de un servidor para entrenar los modelos de Machine Learning, en las tres opciones de plataformas probadas:

RECURSOS	ORACLE - YEAR	AZURE (L32) - YEAR	ON - PREMISE
RAM 256 GB + CPU 32 Cores	19.998 €	26.568 €	14.670 €
Almacenamiento 1TB	490 €		4.220 €
Total	**20.488 €**	**26.568 €**	**18.890 €**

Tabla 8.Comparativa costes en las 3 Plataformas a estudio.

6. Desarrollo Del Proyecto – Smart Pricing

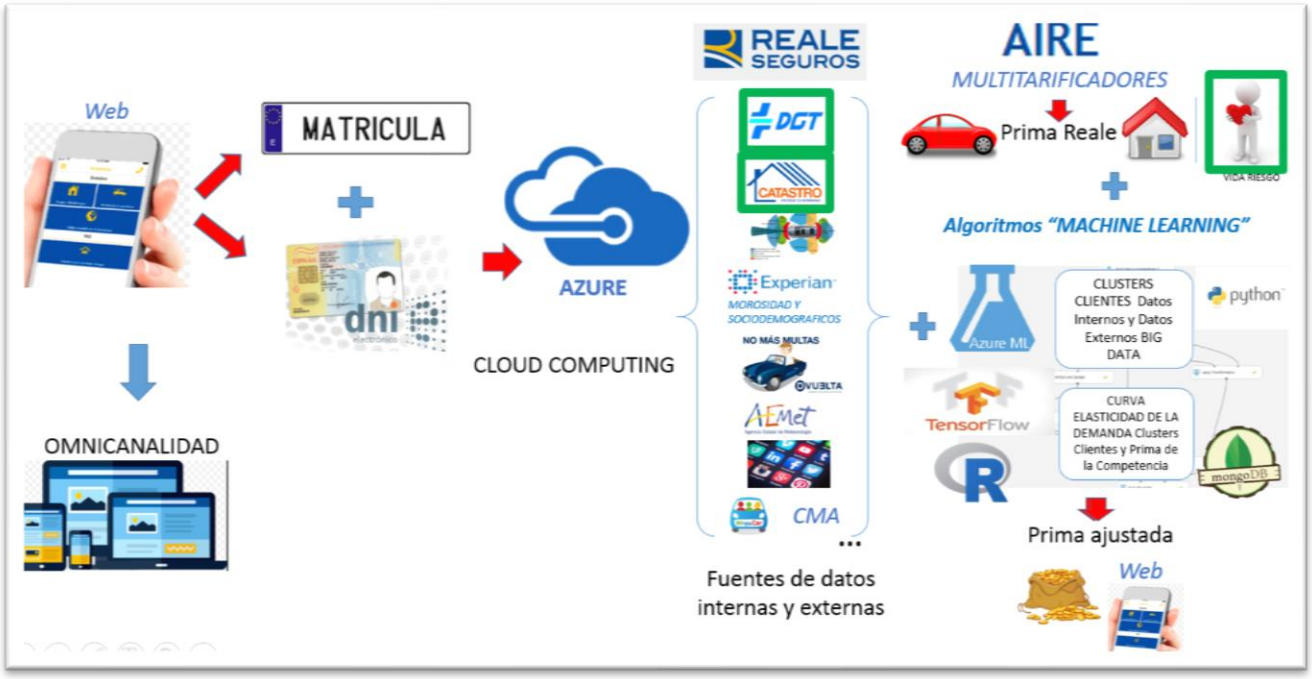


Figura 22. Desarrollo del proceso.

6.1 Comprensión del negocio

Esta fase es probablemente la más importante, ya que contiene la comprensión de los objetivos desde la perspectiva empresa-cliente.

Conociendo bien el problema a resolver, se pueden recoger los datos correctos e interpretar adecuadamente los resultados. En este proyecto, esta fase es muy pequeña ya que los objetivos del proyecto los define la propia empresa, Reale. En nuestro caso son:

- Cliente como usuario: implica la creación de una nueva plataforma visual y actual para la simulación de un seguro.
- Necesidad de optimización del margen que el cliente deja en Reale.

Evaluación de la situación

Como hemos podido comprobar personalmente, la actual plataforma para interactuar con nuestros servicios queda obsoleta teniendo en cuenta los avances que existen hoy en día. Una solicitud que inicialmente parece que va a ser rápida, se convierte en algo tedioso y lento, se piden

numerosos datos y antes de acabar el proceso, muchos lo abandonan, con la consiguiente pérdida de clientes potenciales.

Por otro lado, Reale dispone de un multitarificador propio (AIRE), que es el punto de partida para cualquier proceso de tarificación.

Para ajustar la prima que actualmente ofrece Reale, es necesario incluir en el proceso de tarificación más información, como por ejemplo:

- DGT
- Aemet
- Experian morosidad y sociodemográficos
- Webservice catastro: datos de inmuebles.
- Multas

6.2 Entendimiento de los datos

En esta etapa se han recolectado los datos de las distintas fuentes internas y externas. Para los datos internos se ha tomado como referencia el histórico de clientes desde 2010

Recolección de datos iniciales.

La primera tarea en esta segunda fase del proceso de CRISP- DM, es la recolección de los datos iniciales y su adecuación para el futuro procesamiento. La recolección de datos se realiza directamente a la base de datos Oracle con SQL y los Data Set se crean desde R.

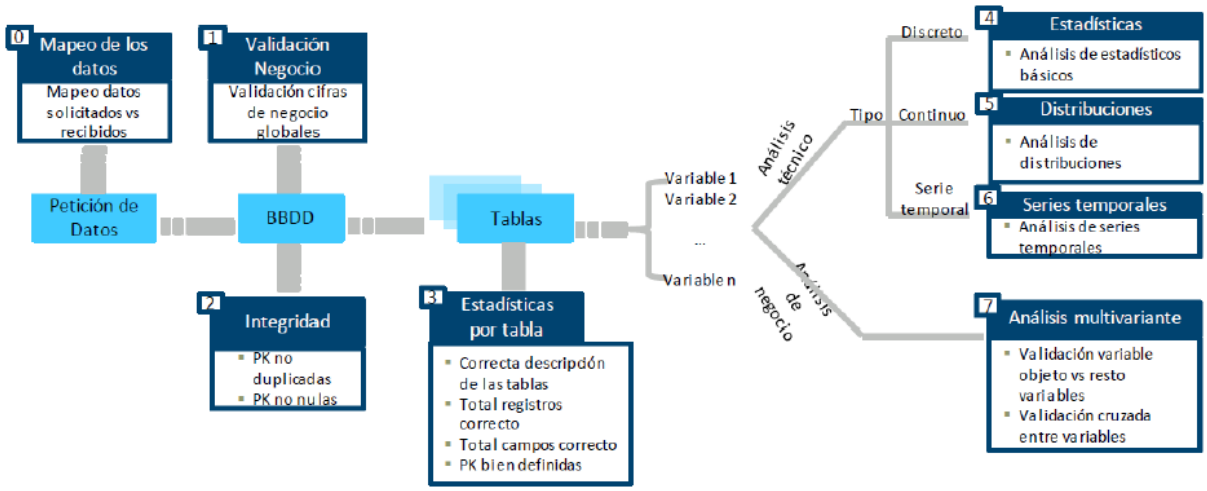


Figura 23. Recolección de Datos

Se recolectan datos de pólizas, siniestralidad, morosidad, sociodemográficas de Experian, meteorológica de AEMET. De estas extracciones iniciales se consigue un data set, que consta de 183 variables y total de 3.458.271 Registros.

Exploración de datos

Se procede a su exploración, cuyo fin es encontrar una estructura general para los datos. Esto involucra la aplicación de pruebas estadísticas básicas, que revelen propiedades en los datos recién adquiridos, se crean tablas de frecuencia y se construyen gráficos de distribución. La salida de esta tarea es un informe de exploración de los datos.

Verificación de la calidad de los datos

Se efectúan verificaciones sobre los datos, para determinar la consistencia de los valores individuales de los campos, la cantidad y distribución de los valores nulos, y para encontrar valores fuera de rango, los cuales pueden constituirse en ruido para el proceso. La idea en este punto, es asegurar la completitud y corrección de los datos.

En la siguiente tabla se reflejan las variables recopiladas, con un ejemplo del tipo de dato que contiene.

Tabla 9. Tabla Variables recopiladas.

VARIABLE	TIPO_VARIABLE	Ejemplo Valor	Ejemplo Valor
CODIGO_CLIENTE	NUMBER(38)	945406	2316127
FECHA_ALTA_CLIENTE	DATE	22/03/2001	02/12/1998
YEAR_ALTA	NUMBER	2001	1998
QUARTER_ALTA	NUMBER	1	4
MONTH_ALTA	NUMBER	3	12
FECHA_BAJA_CLIENTE	DATE		22/09/2012
YEAR_BAJA	NUMBER		2012
QUARTER_BAJA	NUMBER		3
MONTH_BAJA	NUMBER		9
CODIGO_ANTIGUEDAD_CLIENTE	NUMBER(5,2)	16	13
CONT_VIGOR	NUMBER	1	0
YEARS_VIGENTE	NUMBER	16,62	13,8
MONTHS_VIGENTE	NUMBER	199,48	165,65
DAYS_VIGENTE	NUMBER	6073	5043
CODIGO_TRAMO_EDAD	NUMBER(2)	20	20
TRAMO_EDAD	VARCHAR2(30)	>30 y <=60	>30 y <=60
EDAD	NUMBER(4)	43	57
CODIGO_POSTAL	VARCHAR2(5)	8110	15010
COD_PROVINCIA	NUMBER(2)	8	15
PROVINCIA	VARCHAR2(40)	BARCELONA	A CORUÑA

COD_COMU_AUTONOMA	NUMBER(3)	7	12
COMU_AUTONOMA	VARCHAR2(50)	CATALUNYA	GALICIA
CODIGO_CLASIFICACION_EMPLEADO	VARCHAR2(15)	1	1
CLASIFICACION_EMPLEADO	VARCHAR2(15)	NO EMPLEADO	NO EMPLEADO
FECHA_NACIMIENTO	DATE	23/11/1973	01/01/1960
GLOBAL_POLIZAS	NUMBER(6)	1	0
COD_SEGMENTO	NUMBER	140	140
SEGMENTO	VARCHAR2(4000)	MERCADO MASIVO (F¿¿sicos)	MERCADO MASIVO (F¿¿sicos)
SECCIONCENSAL	NUMBER	812501002	1503006019
NOTA1	VARCHAR2(4000)	Z	OTROS
NOTA2V	VARCHAR2(4000)	3	0
SOCSEGMENTOSEGUROS	VARCHAR2(4000)	TA3	TA2
SOCSEGMENTOSEGUROSDESC	VARCHAR2(4000)	Consumidor de Seguros	Gran Consumidores de Seguros
SOCRENTAMEDIA	NUMBER	26228,20602	29838,31
SOCSTATUS	NUMBER	118	94
SOCVIDMUERTE	NUMBER	97	75
SOCVIDMUERVENC	NUMBER	67	65
SOCHIPOT	NUMBER	85	40
SOCCONTCIA	NUMBER	96	157
SOCCONTMEDIADOR	NUMBER	57	131
SOCCONTBANCO	NUMBER	103	48
SOCPOLM25	NUMBER	98	74
SOCPOL2550	NUMBER	49	85
SOCPOL5075	NUMBER	106	117
SOCPOL75100	NUMBER	119	74
SOCPOLM100	NUMBER	45	30
SOCSEGUROMEDICO	NUMBER	115	131
SOCSEGMEDTOT	NUMBER	90	140
SOCSEGMEDPARC	NUMBER	131	122
SOCASEGPAGASEGMED	NUMBER	125	152
SOCSEGMEDEMP	NUMBER	23	59
SOCMOTPRECIO	NUMBER	103	20
SOCMOTOFER	NUMBER	92	137
SOCMOTCONF	NUMBER	120	83
SOCMOTNECE	NUMBER	77	90
SOCMOTSERV	NUMBER	107	160
SOCMOTCLAR	NUMBER	87	100
SOCTER	NUMBER	104	87
SOCTEROPC	NUMBER	64	62
SOCTRCF	NUMBER	182	192
SOCTRSF	NUMBER	38	30
SOCCONTAUTCIA	NUMBER	84	68

SOCCONTAUTMEDIADOR	NUMBER	106	199
SOCCONTAUTBANCO	NUMBER	171	94
SOCMOTRAPI	NUMBER	78	125
SOCCONTAUTTELF	NUMBER	78	43
SOCCONTAUTINT	NUMBER	69	40
SOCCONTAUTPERS	NUMBER	93	126
SOCSIEMPRECIA	NUMBER	83	88
SOCANTMA5	NUMBER	74	97
SOCCOCHES1	NUMBER	112	127
SOCCOCHES2	NUMBER	81	53
SOCCOCHES0	NUMBER	111	264
SOCESTRES	NUMBER	95	84
SOCAGOTAMIENTO	NUMBER	68	96
SOCSINSEGHOGAR	NUMBER	80	108
SOCCONTM5	NUMBER	121	83
SOCCOMPRAINTERNET	NUMBER	55	32
SOCTIPOLOGIA	NUMBER	26	4
SOCGRUPOMOSAIC	VARCHAR2(4000)	F	A
SOCFFAMILIAS	NUMBER	24	75
SOCTIPVIV	NUMBER	68	59
SOCURBANIDAD	NUMBER	86	95
SOCTRANSITORIEDAD	NUMBER	60	73
SOCCALIDADVIV	NUMBER	23	31
SOCFHOGAR	NUMBER	54	83
CUENCA	VARCHAR2(4000)	0	1
TMAX	NUMBER	26,77675382	23,62875011
TMIN	NUMBER	6,249866683	8,117083023
DIAS_TMIN_0	NUMBER	1,511435482	7,30E-08
DIAS_TMIN_5	NUMBER	0,060206673	1,28E-09
DIAS_TMIN_20	NUMBER	0,320920811	0,005
DIAS_TMAX_25	NUMBER	10,28417409	1,747916868
DIAS_TMAX_30	NUMBER	3,873155739	0,150000048
PMES77	NUMBER	480,982273	852,5291783
PMAX77	NUMBER	225,9827178	216,9208354
DP10	NUMBER	4,417634174	10,5541667
DP100	NUMBER	1,511801866	2,762500054
DP300	NUMBER	0,34220474	0,283333343
DLLUVIA	NUMBER	7,757896416	15,03333323
DNIEVE	NUMBER	0,109252591	0,012499999
DGRANIZO	NUMBER	0,151034331	0,758333265
DTORMENTA	NUMBER	1,499372356	1,149999884
DNIEBLA	NUMBER	0,872182257	2,533333183
DESCARCHA	NUMBER	0,512281425	0,166666688

DROCIO	NUMBER	3,189281496	7,52499942
DNIEVESUE	NUMBER	0,036988354	0
R_MAX_VEL	NUMBER	59,55397014	73,91045792
MUNICODE	NUMBER	8125	15030
DENSIDAD	NUMBER	1484,779251	4072,625633
EDADMEDIA	NUMBER	39,65541996	47,26420269
EXT_EXT	NUMBER	0,091363128	0,064140897
HABITANTES	NUMBER	34802	243978
SOCPAROTOT	NUMBER	0,1111	0,1293
SOCPAROHME25	NUMBER	0,124	0,0874
SOCPAROH2545	NUMBER	0,0738	0,1094
SOCPAROHMA45	NUMBER	0,1214	0,1452
SOCPAROMME25	NUMBER	0,1002	0,0821
SOCPAROM2545	NUMBER	0,1063	0,1308
SOCPAROMMA45	NUMBER	0,1495	0,1444
SOCPAROAGRI	NUMBER	0,001	0,0021
SOCPAROINDU	NUMBER	0,0158	0,0097
SOCPAROCONS	NUMBER	0,0112	0,0113
SOCPAROSER	NUMBER	0,0746	0,0951
SOCPAROSINEMP	NUMBER	0,0086	0,0111
MARCA_MOROSIDAD	NUMBER	1	1
MARCA_SOCIODEMO	NUMBER	1	1
MARCA_METEO	NUMBER	1	1
CANTIDAD_POLIZAS	NUMBER	1	1
PRIMA_BRUTA	NUMBER	795,15	1369,027026
DESCUENTO_EMPLEADO	NUMBER	0	0
DESCUENTO_COMERCIAL	NUMBER	-159,03	-126,0377778
DESCUENTO_GENERAL	NUMBER	0	-108,6492486
DESCUENTO_OPERADOR	NUMBER	0	0
PRIMA_ADQUIRIDA	NUMBER	636,12	1134,34
SINIESTRALIDAD	NUMBER	75,98	498,4
MARGEN_BRUTO	NUMBER	719,17	870,6270263
TOTAL_SINIES_DESDE2005	NUMBER	0	2
AUT_TOT_POLIZAS	NUMBER	1	1
AUT_TOT_PRIMA_BRUTA	NUMBER	795,15	1369,027026
AUT_TOT_PRIMA_ADQ	NUMBER	636,12	1134,34
AUT_TOT_SINIESTRALIDAD	NUMBER	75,98	498,4
AUT_TOT_MARGEN_BRUTO	NUMBER	719,17	870,6270263
HOGAR_TOT_POLIZAS	NUMBER	0	0
HOGAR_TOT_PRIMA_BRUTA	NUMBER	0	0
HOGAR_TOT_PRIMA_ADQ	NUMBER	0	0
HOGAR_TOT_SINIESTRALIDAD	NUMBER	0	0
HOGAR_TOT_MARGEN_BRUTO	NUMBER	0	0

COMER_TOT_POLIZAS	NUMBER	0	0
COMER_TOT_PRIMA_BRUTA	NUMBER	0	0
COMER_TOT_PRIMA_ADQ	NUMBER	0	0
COMER_TOT_SINIESTRALIDAD	NUMBER	0	0
COMER_TOT_MARGEN_BRUTO	NUMBER	0	0
ACCI_TOT_POLIZAS	NUMBER	0	0
ACCI_TOT_PRIMA_BRUTA	NUMBER	0	0
ACCI_TOT_PRIMA_ADQ	NUMBER	0	0
ACCI_TOT_SINIESTRALIDAD	NUMBER	0	0
ACCI_TOT_MARGEN_BRUTO	NUMBER	0	0
RC_TOT_POLIZAS	NUMBER	0	0
RC_TOT_PRIMA_BRUTA	NUMBER	0	0
RC_TOT_PRIMA_ADQ	NUMBER	0	0
RC_TOT_SINIESTRALIDAD	NUMBER	0	0
RC_TOT_MARGEN_BRUTO	NUMBER	0	0
COMUN_TOT_POLIZAS	NUMBER	0	0
COMUN_TOT_PRIMA_BRUTA	NUMBER	0	0
COMUN_TOT_PRIMA_ADQ	NUMBER	0	0
COMUN_TOT_SINIESTRALIDAD	NUMBER	0	0
COMUN_TOT_MARGEN_BRUTO	NUMBER	0	0
INDUST_TOT_POLIZAS	NUMBER	0	0
INDUST_TOT_PRIMA_BRUTA	NUMBER	0	0
INDUST_TOT_PRIMA_ADQ	NUMBER	0	0
INDUST_TOT_SINIESTRALIDAD	NUMBER	0	0
INDUST_TOT_MARGEN_BRUTO	NUMBER	0	0
RIES_TOT_POLIZAS	NUMBER	0	0
RIES_TOT_PRIMA_BRUTA	NUMBER	0	0
RIES_TOT_PRIMA_ADQ	NUMBER	0	0
RIES_TOT_SINIESTRALIDAD	NUMBER	0	0
RIES_TOT_MARGEN_BRUTO	NUMBER	0	0
RESTO_TOT_POLIZAS	NUMBER	0	0
RESTO_TOT_PRIMA_BRUTA	NUMBER	0	0
RESTO_TOT_PRIMA_ADQ	NUMBER	0	0
RESTO_TOT_SINIESTRALIDAD	NUMBER	0	0
RESTO_TOT_MARGEN_BRUTO	NUMBER	0	0

Adicionalmente para cada una de las variables, se analizó cual era la distribución de la información y se realizó un top 10 de los valores con mayor frecuencia, por cada una de las variables.

A continuación, sólo vamos a comentar algunas de las variables:

CONT_VIGOR: Esta variable indica si el cliente está o no en vigor en la actualidad y podemos observar que según el histórico desde 2010, un 48% ya no es cliente.

Tabla 10. Variable CONT_VIGOR

	CONT_VIGOR	COUNT(1)	%
1	1	1.777.038	51,39%
2	0	1.681.233	48,61%
	TOTAL CLIENTES	**3.458.271**	

SOCSEGMENTOSEGUROS		COU
TA2	Gran Consumidores de Seguros	
TA1	Siempre Asegurados con Cobertura Total	
TA3	Consumidor de Seguros	
TA0	Sin definir	
TA8	Consumidor Solo Auto	
TA6	Seguros Obligatorios	
TA5	Seguros Obligatorios + Salud	
TA9	Consumidores Espor¿¿dicos	
TA4	Consumidor de Seguros M¿¿xima Cobertura	
TA4	Consumidor de Seguros M¿xima Cobertura	
TA9	Consumidores Espor¿dicos	
TA7	Consumidor Ocasional	

	YEAR_ALTA	COUNT(1)
1	2013	259.294
2	2014	250.149
3	2015	246.276
4	2012	243.801
5	2009	243.007
6	2010	227.279
7	2016	219.038
8	2011	194.327
9	2008	181.920
10	2017	**171.290**
11	2007	130.450

	YEAR_BAJA	COUNT(1)
1	VIGENTES	1.777.038
2	2016	241.455
3	2015	230.169
4	2014	222.428
5	2013	222.252
6	2012	211.992
7	2017	**189.714**
8	2011	187.831
9	2010	173.215
10	2018	2.177

Antigüedad Clientes de Baja			
	MONTHS_VIGENTE	COUNT	% Abandono
1	12	224454	99,92%
2	24	184887	99,66%
3	36	129037	99,54%
4	48	96330	99,49%
5	0	78304	100,00%
6	60	70843	99,43%
7	72	54500	99,88%
8	84	43277	99,76%
9	96	33402	98,93%

º	**CLIENTES EN VIGOR**		
	COD_PROVINCIA		COUNT(1)
1	8	BARCELONA	174.763
2	46	VALENCIA	115.757
3	28	MADRID	113.051
4	15	A CORUÑA	110.881
5	43	TARRAGONA	63.915
6	3	ALACANT	63.640
7	50	ZARAGOZA	59.408

10	108	26623	99,05%
11	120	22217	99,04%
12	132	19217	

8	30	MURCIA	55.889
9	29	MALAGA	49.605
10	41	SEVILLA	46.658

6.3 Preparación de los datos

Se realizan de las primeras extracciones con información de datos de clientes de siniestralidad.

La fase de preparación de datos engloba todas las actividades necesarias para construir el conjunto de datos final, que será usado en la fase de modelado a partir de los datos iniciales. Estas tareas de preparación de datos van a ser ejecutadas repetidas veces y no pueden realizarse en cualquier orden. En general incluyen la selección y transformación de tablas, registros y atributos y limpieza de datos.

Estructuración de los datos

Incluye las operaciones de preparación de los datos tales como la generación de nuevos atributos a partir de atributos ya existentes, integración de nuevos registros o transformación de valores para atributos existentes.

Formateo de los datos

Realización de transformaciones de los datos sin modificar su significado, esto, con la idea de permitir o facilitar el empleo de alguna técnica de DM en particular, como por ejemplo la reordenación de los campos y/o registros de la tabla o el ajuste de los valores de los campos a las limitaciones de las herramientas de modelación (eliminar comas, tabuladores, caracteres especiales, máximos y mínimos para las cadenas de caracteres, etc.). En esta etapa, el formateo más complejo que nos encontramos fue el de separador de decimales, ya que en la importación de un mismo data set, había variables numéricas que tenían un separador de "." y otras con la "," y por tanto fue necesario hacer muchas transformaciones de estos campos incluso pasándolos por carácter, hasta conseguir tenerlos unificados.

Limpieza de los datos.

Inicialmente se probó a ejecutar algoritmos de los paquetes de R, como mice y caret para llenar los campo nulos, pero dado el volumen de datos que se gestionaba, estos algoritmos fueron muy lentos en su ejecución, algunos de ellos tardaron hasta 10 días, sin conseguir un claro patrón de llenado. Al final optamos por eliminar los registros que tenían valores nulos para el entrenamiento de los modelos y el data set quedo con más de 2 millones y medio de registros.

Matriz de correlaciones.

La matriz de correlaciones se utilizó para empezar a hacer una reducción de las variables numéricas. En la siguiente gráfica se puede observar una gráfica de una de las matrices de correlaciones trabajadas, dada la volumetría de variables, pue un poco complicado de analizar.

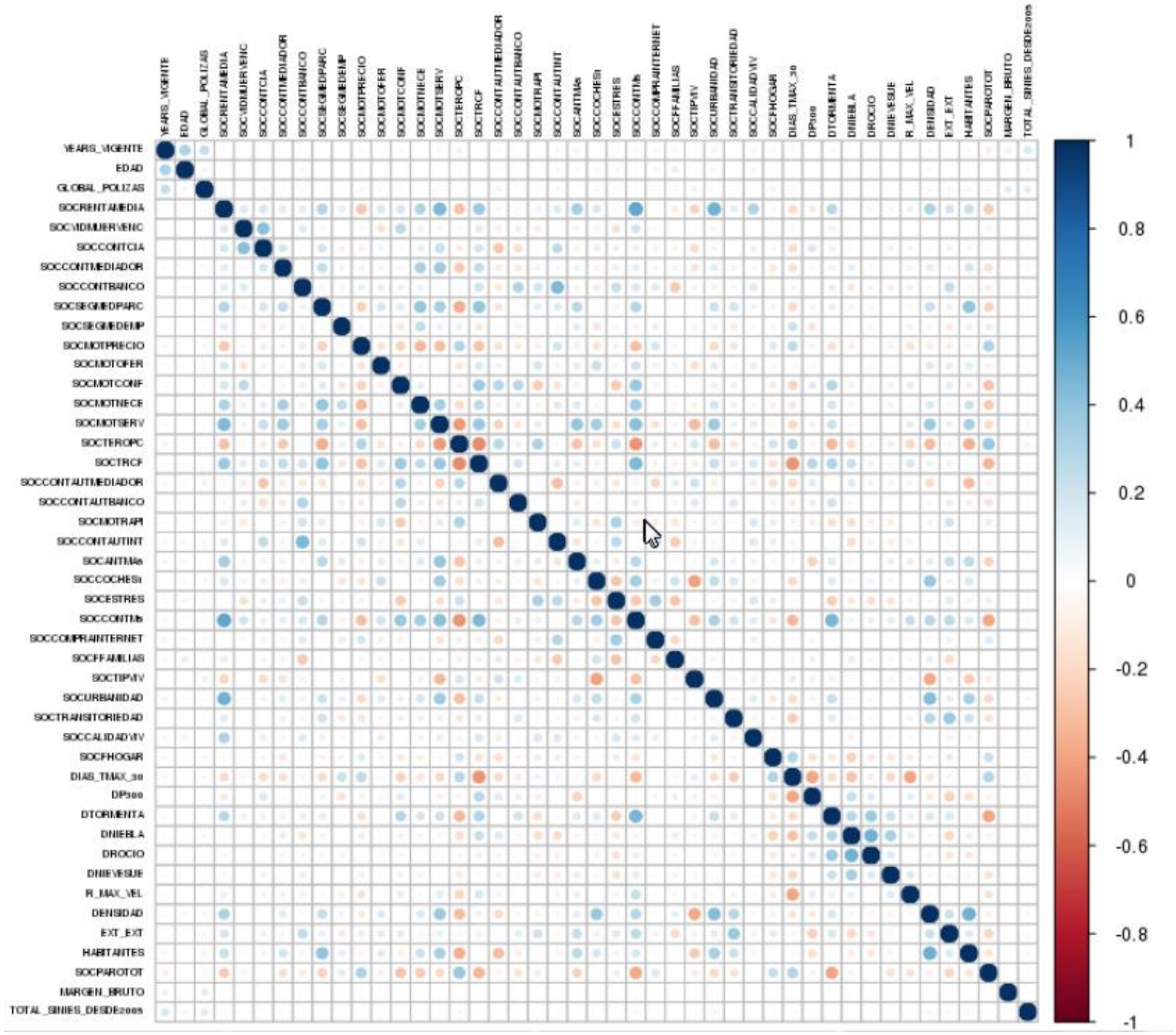


Figura 24. Matriz de Correlación

Variables Significativas

Para continuar con una reducción de variables, se empezó a ejecutar algoritmos de Auto Machine Learning de H2O. ia y al final del proceso, de los modelos liner analizamos las variables más significativas para los modelos y poco a poco fuimos haciendo una reducción de variables, hasta llegar a tener una lista de sólo 24 variables. En el Anexo H, se puede observar el log de alguna de esas ejecuciones.

Para cada una de las variables finales, generamos una gráfica en la cual se incluye información estadística de la misma y su distribución mediante un histograma y con una representación en línea se incluye la Media del MARGEN_BRUTO, como segunda variable; damos importancia a la variable MARGEN_BRUTO dado que es la variable Target en los modelos de Machine Learning trabajados.

A continuación, en este apartado incluimos solo la gráfica de la variable CANTIDAD_POLIZAS y en el Anexo "E" del documento se incluyen el resto de las gráficas.

Gráfica 1. Variable CANTIDAD_POLIZAS

En la siguiente tabla, las variables utilizadas en los modelos son las se encuentran en la siguiente tabla:

Variable	Descripción	Tipo
MARGEN_BRUTO	Margen bruto beneficio	double
PRIMA_BRUTA	Prima bruta	double
TOTAL_SINIES_DESDE2005	Cantidad siniestros	Integer
CON_VIGOR	Indicador cliente en vigor en la actualidad	Integer
YEARS_VIGENTE1	Años vigencia cliente	Integer
CANTIDAD_POLIZAS	Número de pólizas contratadas	Integer
RAMOS	Cantidad ramos diferentes que tiene contratados (Auto, hogar, Diversos, etc)	Integer
EDAD	Edad	Integer
NOTA1	Indicador asociado a la morosidad del cliente	Factor
NOT2V	Indicador de Morosidad	Factor
COD_SEGMENTO1	Clasificación por segmentos definidos por Reale	Factor
SOCSEGMENTOSEGUROS	Clasificación de segmentos definido por los datos sociodemográficos	Factor
SOCGRUPOMOSAIC	Agrupación de segmentos definida por Experian	Factor
SOCRENTAMEDIA	Renta media	double
SOCURBANIDAD	Indice de urbanidad.	double

SOCPAROTOT	Indice de paro	double
COD_PROVINCIA	Código Provincia	Factor
HABITANTES	Número de habitantes	Integer
DENSIDAD	Densidad de población por código postal	double
TMIN	Promedio temperatura minima anual por código postal	double
R_MAX_VEL	Promedio anual de rachas de viento	double
DNIEBLA	Dias de niebla en la zona	double
DTORMENTA	Días de tormenta	double
DIAS_TMAX_30	Promedio anual días temperatura <30º registradas por código postal	double
DP100	Días con precipitaciones superiores a los 100mm por código postal	double
CLASIFICACION_EMPLEADO	Clasificación de la relación con Empleados de Reale	Factor

Tabla 11. Variables utilizadas en los modelos.

6.4 Modelización y Evaluación

Para el proceso de optimización del precio de la Prima, uno de los objetivos principales de este proyecto, planteamos el desarrollo de varios algoritmos de machine learning, que se pueden observar en la siguiente gráfica donde se plasma el flujo completo del proyecto.

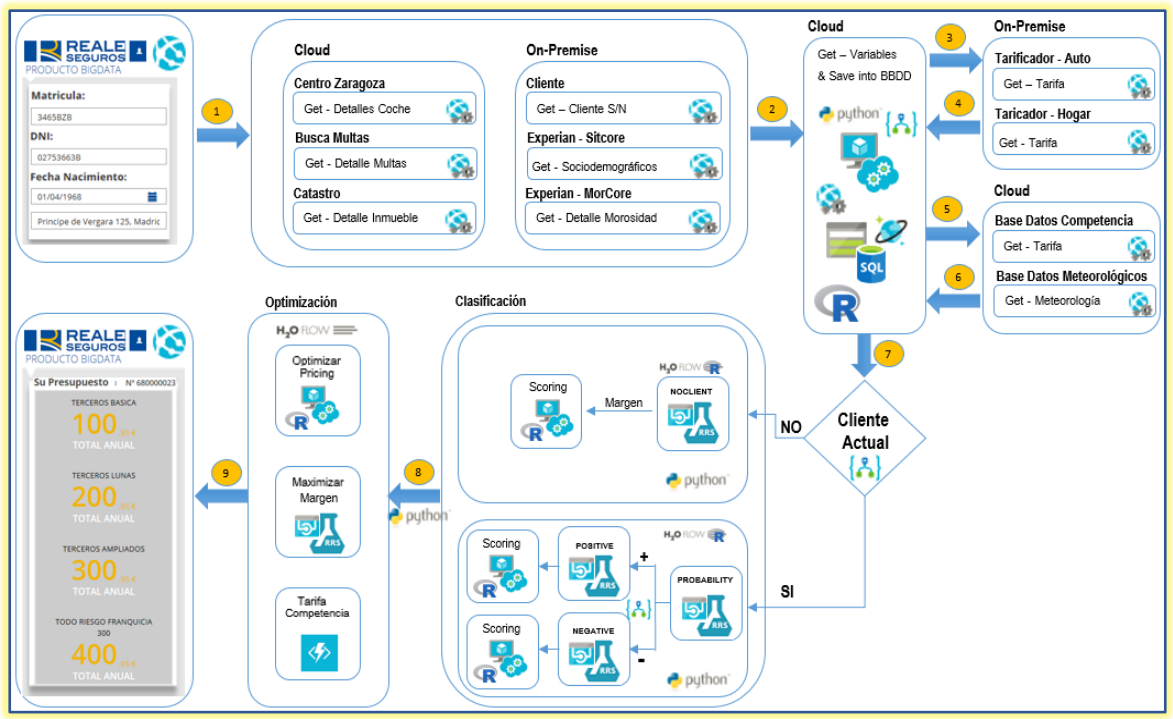


Figura 25. Diagrama de Flujo del Proyecto

En el diagrama de flujo del proyecto, se pueden identificar los cuatro algoritmos desarrollados para permitir la clasificación de los clientes para posteriormente aplicar un proceso de optimización del precio de la prima.

Como se puede observar en el anterior diagrama, una vez registrados los datos básicos para la tarificación, entre los primeros procesos que se ejecutan esta la verificación de si es cliente de Reale y resultado de esta comprobación es el que determina los algoritmos de Machine Learning que se ejecutarán, con la siguiente secuencia:

- **Sí** es cliente de Reale, lo primero que ejecutamos es un algoritmo Binomial llamado **PROBABILITY** que va a predecir sí el cliente va a dejar un Margen Positivo o Negativo, retornando un 1 en caso positivo o un 0 en caso negativo y a continuación según el resultado se invocarán los siguientes algoritmos de Regresión que van a predecir el Margén:
    - Si resultado es 1, se invoca el algoritmo **POSITIVE**
    - SI resultado es 0, se invoca el algoritmo **NEGATIVE**

- Cuando **No** es cliente de Reale, solo se ejecuta el algoritmo de regresión que hemos denominado **NOCLIENT**, que va a predecir el Margen que dejará el cliente a Reale.

Siguiendo la metodología esta etapa del proyecto se basa y completa mediante la correcta consecución de las siguientes fases:

• Selección de la técnica de modelado. Esta tarea consiste en la selección de la técnica más apropiada al tipo de problema a resolver. Para esta selección, se debe considerar el objetivo principal del proyecto y la relación con las herramientas de existentes. Por ejemplo, si el problema es de clasificación, se podrá elegir de entre árboles de decisión…

• Generación del plan de prueba. Procedimiento destinado a probar la calidad y validez del mismo. Por ejemplo, en una tarea supervisada típicamente se separan los datos en dos conjuntos, uno de entrenamiento y otro de prueba, para luego construir el modelo basado en el conjunto de entrenamiento y medir la calidad del modelo generado con el conjunto de prueba

• Construcción del Modelo. Después de seleccionada la técnica, se ejecuta sobre los datos previamente preparados para generar uno o más modelos. Todas las técnicas de modelado tienen un conjunto de parámetros que determinan las características del modelo a generar. La selección de los mejores parámetros es un proceso iterativo y se basa exclusivamente en los resultados generados. Estos deben ser interpretados y su rendimiento justificado.

• Evaluación del modelo. Se interpretan los modelos, de acuerdo al conocimiento preexistente del dominio y los criterios de éxito preestablecidos. Expertos en el dominio del problema juzgan los modelos dentro del contexto del dominio y expertos en Data Mining aplican sus propios criterios (seguridad del conjunto de prueba, perdida o ganancia de tablas, etc...).

6.4.1 Modelos Machine Learning

Después de muchas investigaciones y pruebas de rendimientos, optamos por utilizar la interfaz analítica de cloud computing H2o ai, la cual describimos a continuación por el nivel de importancia que ha tenido en el desarrollo de este proyecto.

H2O ai

H2O, Es una herramienta de código abierto para análisis de Big Data. Los enormes conjuntos de datos que se obtienen a través de Big Data pueden llegar ser demasiado grandes como para ser analizados usando herramientas tradicionales como R. H2O provee estructuras de datos y métodos más afines al tamaño de los datos que se manejan en big data, ya que permite analizar y visualizar grandes conjuntos de datos evitando usar la estrategia tradicional de estudiar solo un subconjunto pequeño de los datos con un paquete estadístico.

Por medio de H2O se pueden ajustar miles de modelos potenciales en la búsqueda de patrones en datos, usando métodos iterativos que proveen respuestas rápidas usando todo el conjunto de datos. En su aproximación a deep learning H2O divide los datos en subconjuntos que se analizan simultáneamente con el mismo método.

Los algoritmos estadísticos de H2O incluyen : Modelos lineales generalizados, gradient boosting, analisis de componentes principales, K-means, bosques aleatorios distribuidos, clasificador bayesiano ingenuo entre otros.

H2O tiene interfaces con Java, R, Python y Scala

Adicionalmente para el despliegue en Producción esta interface permite exportar el modelo final en formatos POJO (Plain Old Java Object) y MOJO (Model Object Java Optimized).

En la fase de entrenamiento de los modelos utilizamos la opción de Auto Machine Learning, para que se generasen diferentes opciones de modelos y luego nos quedábamos con el mejor modelo dentro de los Supervisados, ya que es requisito en Pricing que el modelo sea explicativo, por esta restricción y en búsqueda de algoritmos explicativos, enfocados al análisis de regresión, los modelos seleccionados fueron GBM (Gradient Boosting Machine).

GBM (Gradient Boosting Machine)

Gradient Boosting es una tecnología de machine learning para análisis de regresión y problemas de clasificación que depende de 3 elementos:

- Una función de pérdida que debe ser optimizada.
- Un predictor débil, típicamente un árbol de decisión
- Un modelo aditivo para sumar los predictores débiles que minimizan la función de pérdida.

El objetivo es minimizar la pérdida agregando predictores. Cada vez que se agrega un predictor se optimiza dejando los demás predictores estáticos (sin cambio).

La función de perdida depende del tipo de problema a resolver, por ejemplo una regresión puede usar error cuadrado y una clasificación puede usar perdida logarítmica.

Los predictores débiles son arboles de decisión, típicamente se usan arboles de regresión cuyo resultado después se puede agregar para corregir los residuales en las predicciones.

Los árboles se suelen construir a dos niveles aunque se puede llegar a entre 6 y 8 niveles. El número de niveles se limita a un máximo para hacer que se mantengan débiles.

El modelo aditivo hace que se vayan agregando nuevos árboles manteniendo los anteriores sin cambios. Un procedimiento de potenciación de gradiente se usa para agregar arboles minimizando la perdida.

Como ya hemos comentado, en las diferentes pruebas que se realizaron a lo largo del proyecto donde se partía de cerca de 190 Variables, se fueron realizando análisis de ajustes de variables, revisadas y aprobadas siempre por el departamento técnico actuarial de Reale que tienen a su cargo el desarrollo de los modelos de Pricing.

Antes de entrar en el detalle de cada uno de los modelos de machine learning, vamos a hacer referencia a los principales parámetros estadísticos que utilizamos para parametrizar y evaluar la ejecución de los modelos.

MSE

En un modelo estadístico puede haber un parámetro desconocido para el que debe usarse un estimador. Por medio de MSE (Error Cuadrático Medio) se puede evaluar la diferencia entre un la estimación o predicción y el valor real observado.

El MSE equivale a la suma de la varianza y la desviación al cuadrado del estimador

El MSE es un criterio para seleccionar un estimador apropiado y se usa para determinar la medida en la que el modelo no se ajusta a la información, o si el quitar ciertos términos puede simplificar el modelo de manera beneficiosa. El MSE proporciona una forma para elegir el mejor estimador.

Tener un MSE de cero (0) es ideal pero no es posible en la mayoría de las situaciones. Un MSE de 0 significa que el estimador predice las observaciones con una precisión perfecta.

RMSE

La Raíz del Error Cuadrático Medio es la raíz cuadrada del promedio del error cuadrático entre predicción y observación.

Debido a que los errores se llevan al cuadrado antes de obtener el promedio, el RMSE concede mayor peso a los errores grandes, en consecuencia el RMSE es más útil en los casos en que los errores significativos no son deseables.

AUC

AUC o Area Bajo la Curva es una métrica de clasificación binaria. En una clasificación binaria el resultado es, por ejemplo, 0 o 1.

Es ideal poder medir cual es el nivel de certeza de dicha clasificación. Es decir, cual es el umbral en el que un 0 se convierte en 1.

Por medio de AUC se consideran todos los posibles umbrales para encontrar el punto en que se minimizan los falsos positivos y se maximizan los verdaderos positivos.

Función de distribución Gaussiana

La Distribución Normal o de Gauss es una función de densidad de probabilidad que responde a una curva con forma de campana, con eje de simetría en el punto correspondiente al promedio del universo μ.

La forma de la curva de la distribución depende de sus dos parámetros: la media y la desviación estándar.

La media indica la posición de la campana a lo largo del eje x.

A mayor desviación la curva será más "plana", dado que la distribución, en este caso, presenta una mayor variabilidad. La curva es simétrica respecto a la media.

La importancia de esta distribución radica en que permite modelar numerosos fenómenos naturales, sociales y psicológicos.

Función de distribución Gamma

La distribución Gamma es una distribución adecuada para modelizar el comportamiento de variables aleatorias continuas con asimetría positiva. Es decir, variables que presentan una mayor densidad de sucesos a la izquierda de la media que a la derecha.

Se suele utilizar cuando se trata de representar probabilidades en casos como: número de individuos involucrados en accidentes de tráfico en el área urbana, Altura a la que se inician las precipitaciones, líneas de espera o Ingresos familiares entre otros.

Función de distribución Bernoulli

Se usa la distribución de Bernoulli cuando un proceso aleatorio tiene exactamente dos resultados: evento o no evento y la probabilidad de cada suceso es igual en cualquier realización del experimento.

Las variables de Bernoulli pueden tomar dos valores numéricos, 0 y 1, donde 1 corresponde a un evento y 0 corresponde a un no evento.

Selección aleatoria de elementos de un conjunto

Lanzamiento de una moneda

En el campo de la calidad, clasificar productos terminados (normal, defectuoso)

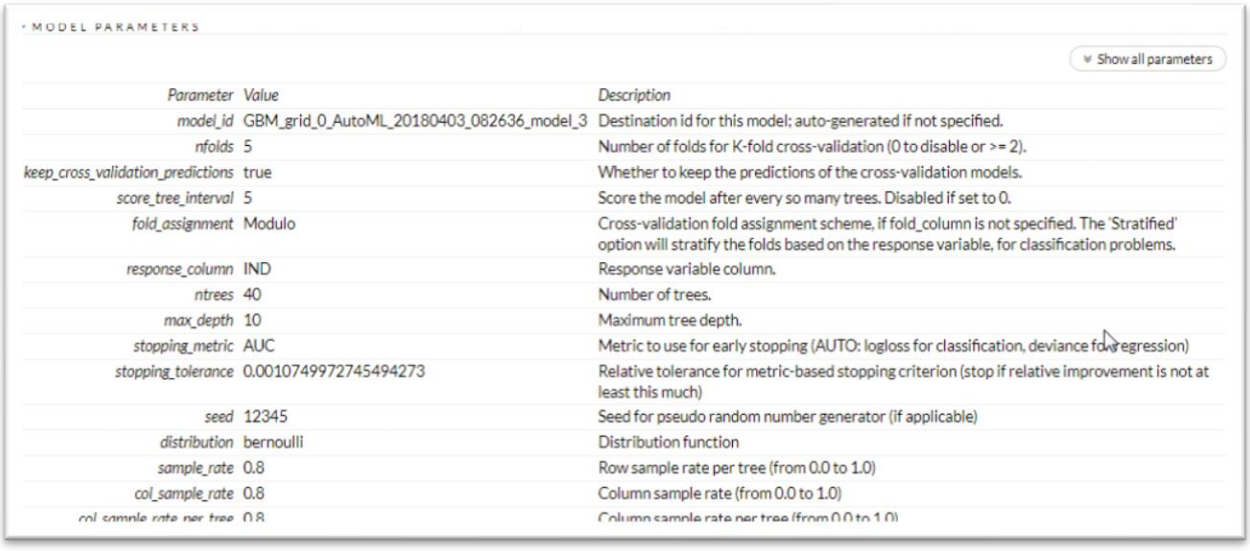
A continuación, se detallan cada uno de los modelos con el conjunto de variables final utilizado para su entrenamiento y en el Anexo H se adjunta, como comentábamos al principio, diferentes modelos trabajados durante los meses de desarrollo del proyecto.

6.4.1.1 Modelo 1. PROBABILITY

Este algoritmo binomial se ha entrenado con la base de datos de clientes de Reale de los últimos 8 años que contiene tanto clientes vigentes como no vigentes, y tiene como propósito, predecir si el margen que dejara el cliente es positivo o negativo.

La métrica utilizada para el stop del Auto ML, ha sido el AUC.

Parámetros:



MODEL PARAMETERS

Show all parameters

Parameter	Value	Description
model_id	GBM_grid_0_AutoML_20180403_082636_model_3	Destination id for this model; auto-generated if not specified.
nfolds	5	Number of folds for K-fold cross-validation (0 to disable or >= 2).
keep_cross_validation_predictions	true	Whether to keep the predictions of the cross-validation models.
score_tree_interval	5	Score the model after every so many trees. Disabled if set to 0.
fold_assignment	Modulo	Cross-validation fold assignment scheme, if fold_column is not specified. The 'Stratified' option will stratify the folds based on the response variable, for classification problems.
response_column	IND	Response variable column.
ntrees	40	Number of trees.
max_depth	10	Maximum tree depth.
stopping_metric	AUC	Metric to use for early stopping (AUTO: logloss for classification, deviance for regression)
stopping_tolerance	0.0010749972745494273	Relative tolerance for metric-based stopping criterion (stop if relative improvement is not at least this much)
seed	12345	Seed for pseudo random number generator (if applicable)
distribution	bernoulli	Distribution function
sample_rate	0.8	Row sample rate per tree (from 0.0 to 1.0)
col_sample_rate	0.8	Column sample rate (from 0.0 to 1.0)
col_sample_rate_per_tree	0.8	Column sample rate per tree (from 0.0 to 1.0)

Figura 26. Parámetros Modelo PROBABILITY

Scoring LogLoss:

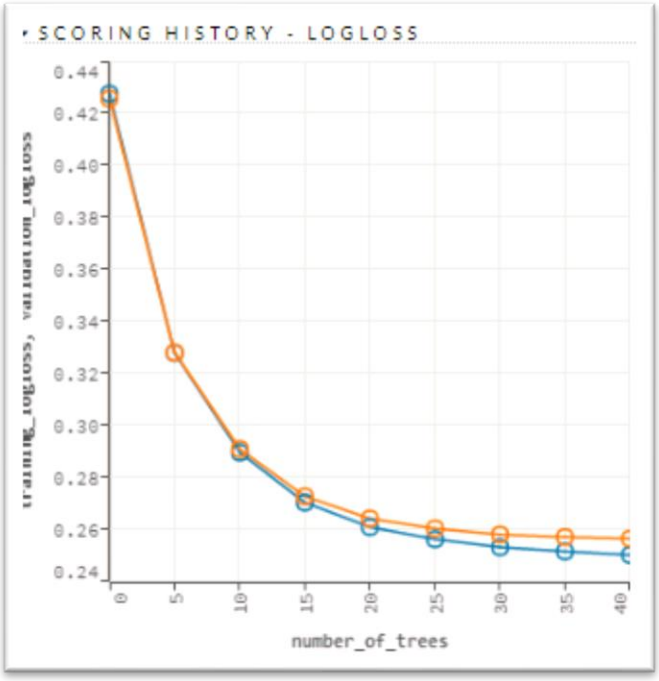


Figura 27. Scoring Modelo PROBABILITY

En el data set de Entrenamiento, se consigue un AUC del 0.9119

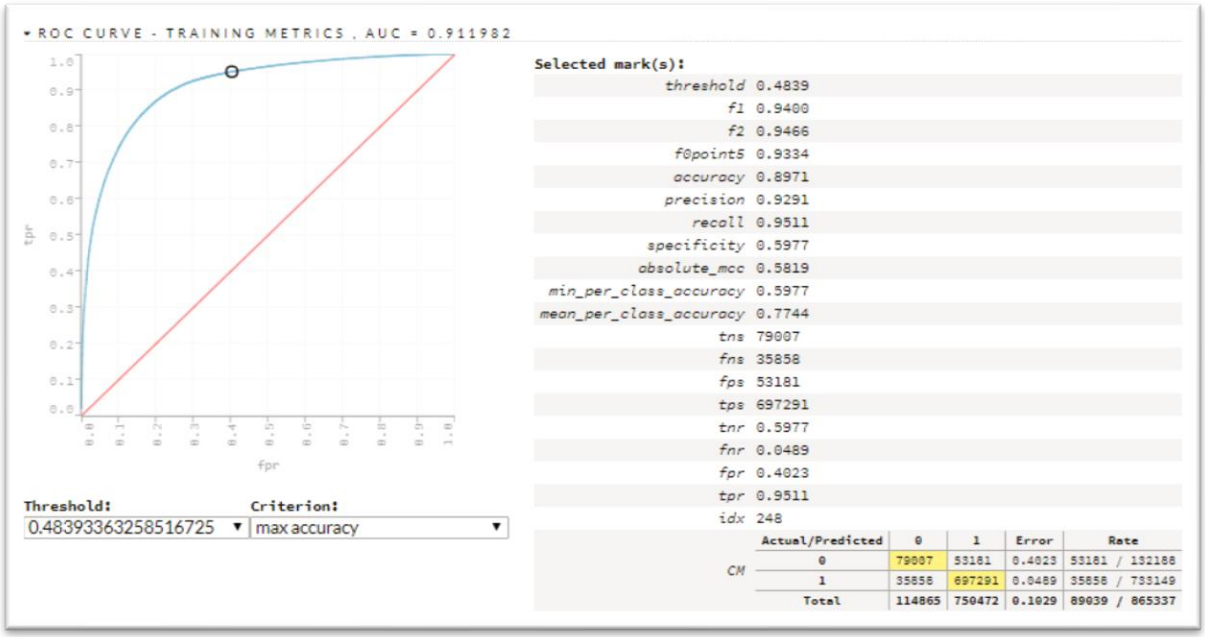


Figura 28. Dataset Entrenamiento Modelo PROBABILITY

Matriz de confusión

	Actual/Predicted	0	1	Error	Rate
CM	0	79007	53181	0.4023	53181 / 132188
	1	35858	697291	0.0489	35858 / 733149
	Total	114865	750472	0.1029	89039 / 865337

Figura 29. Matriz de confusión Dataset Entrenamiento Modelo PROBABILITY

En el data set de validación, se consigue un AUC del 0.9058

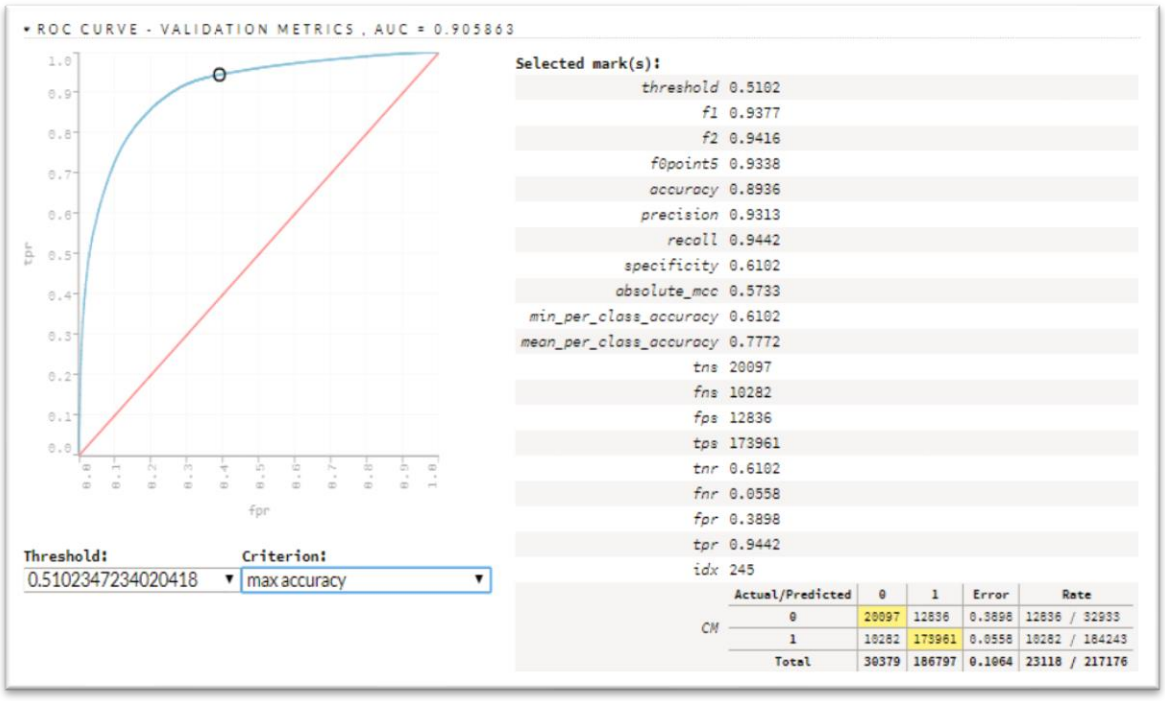


Figura 30. Dataset Validación Modelo PROBABILITY

Matriz de confusión

	Actual/Predicted	0	1	Error	Rate
CM	0	20097	12836	0.3898	12836 / 32933
	1	10282	173961	0.0558	10282 / 184243
	Total	30379	186797	0.1064	23118 / 217176

Figura 31. Matriz de confusión Dataset Validación Modelo PROBABILITY

En la fase de cross validation el AUC es del 0.9061

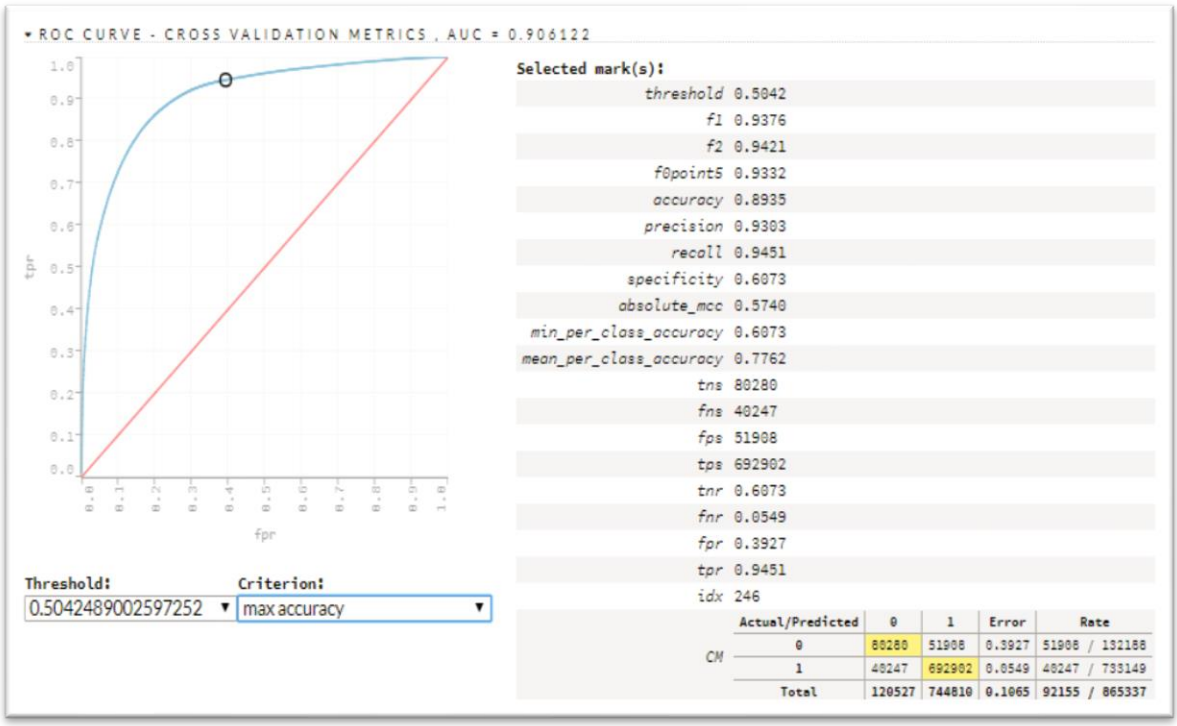


Figura 32. Fase Cross validation modelo PROBABILITY

Matriz de confusión

CM

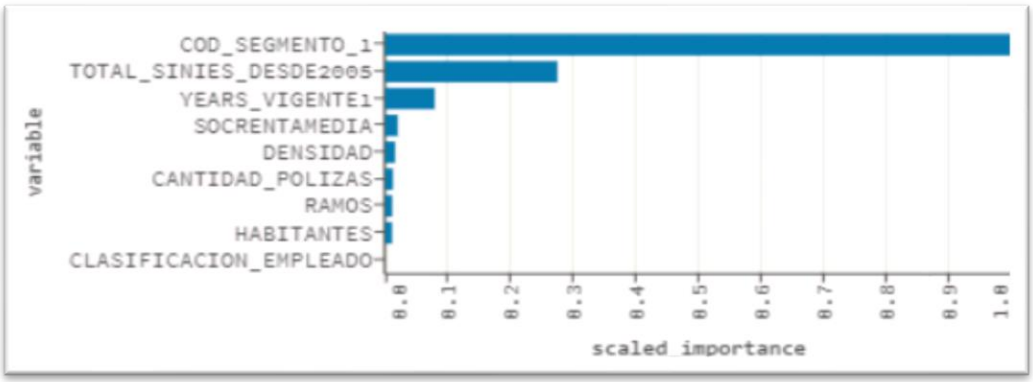
Actual/Predicted	0	1	Error	Rate
0	80280	51908	0.3927	51908 / 132188
1	40247	692902	0.0549	40247 / 733149
Total	120527	744810	0.1065	92155 / 865337

Figura 33. Matriz de confusión fase Cross validation Modelo PROBABILITY

Variables de Importancia

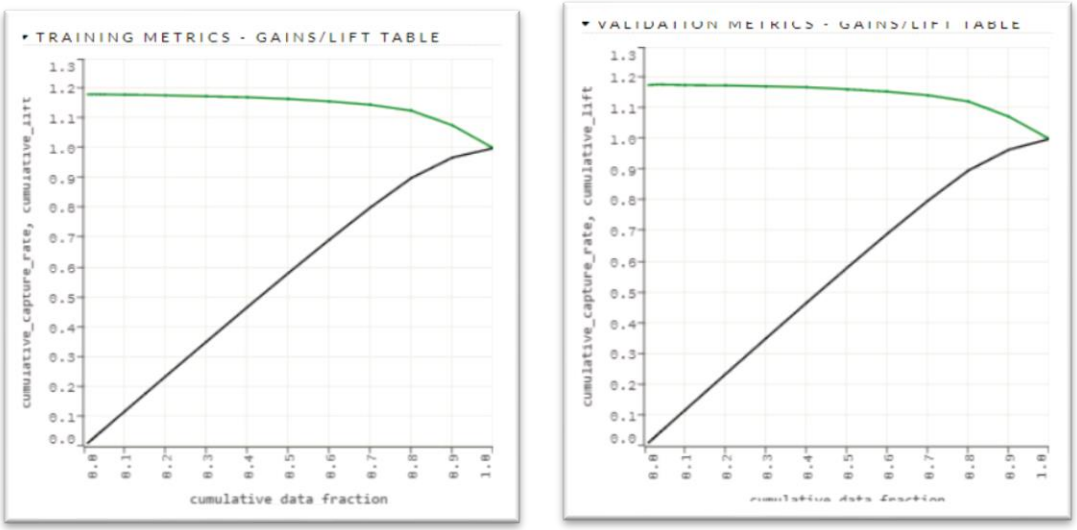
variable	relative_importance	scaled_importance	percentage
COD_SEGMENTO_1	145580.4844	1.0	0.7019
TOTAL_SINIES_DESDE2005	39922.6719	0.2742	0.1925
YEARS_VIGENTE1	11501.9482	0.0790	0.0555
SOCRENTAMEDIA	2957.1948	0.0203	0.0143
DENSIDAD	2337.2861	0.0161	0.0113
CANTIDAD_POLIZAS	1826.7805	0.0125	0.0088
RAMOS	1647.2136	0.0113	0.0079
HABITANTES	1622.6228	0.0111	0.0078
CLASIFICACION_EMPLEADO	15.9867	0.0001	0.0001

Tabla 12. Importancia de Variables modelo Probability



Gráfica 2. Importancia de variables modelo PROBABILITY

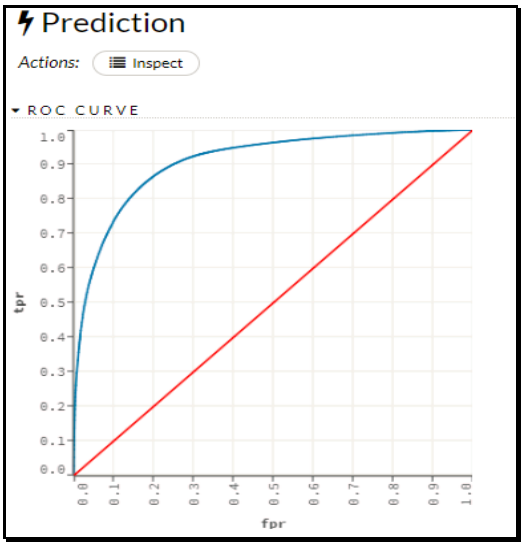
Curvas de Ganancia:



Gráfica 3. Curvas de ganancia Modelo PROBABILITY

Predicciones y Evaluación:

Pasamos las predicciones al set de datos completo y obtenemos los siguientes resultados:



Gráfica 4. Predicciones modelo PROBABILITY

PREDICTION

model	GBM_grid_0_AutoML_20180403_082636_model_3
model_checksum	-2691136673820417024
frame	DATA_VAR_PROBABILIDADES_FULL.hex
frame_checksum	456942757992809856
description	·
model_category	Binomial
scoring_time	1523039181995
predictions	prediction-1d37146a-4f7b-4618-a1b0-fb9df8252e78
MSE	0.076382
RMSE	0.276374
nobs	1804792
custom_metric_name	·
custom_metric_value	0
r2	0.409693
logloss	0.253394
AUC	0.909358
Gini	0.818716
mean_per_class_error	0.236576

PREDICTION - MAXIMUM METRICS

metric	threshold	value	idx
max f1	0.4736	0.9389	259
max f2	0.2344	0.9663	358
max f0point5	0.7612	0.9416	166
max accuracy	0.4965	0.8952	250
max precision	0.9938	0.9996	0
max recall	0.0658	1.0	399
max specificity	0.9938	1.0	0
max absolute_mcc	0.6703	0.5956	199
max min_per_class_accuracy	0.8648	0.8338	116
max mean_per_class_accuracy	0.8492	0.8341	125

Tabla 13. Predicciones Modelo PROBABILITY

Como resultado final en las predicciones se obtiene un **AUC** del **90.93%** y dadas las características del proyecto este resultado es considerado como bueno.

6.4.1.2 Modelo 2 NOCLIENT

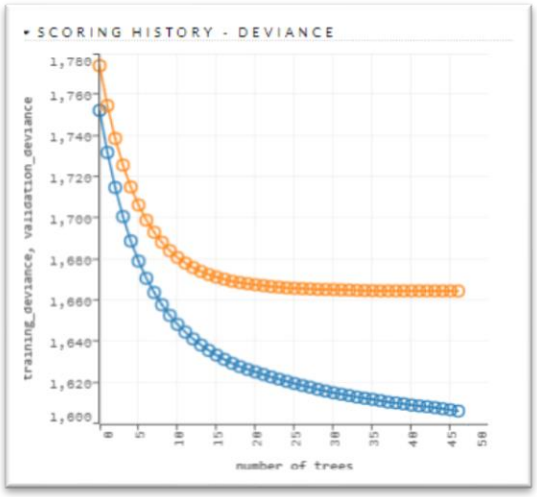
Cuando la comprobación de los datos introducidos, indica que la persona no es cliente actual de Reale o no ha sido cliente en el pasado, ejecutamos este algoritmo para predecir cuál será el margen que dejara el cliente.

Utilizando el método de auto machine learning, se realizaron muchas ejecuciones con diferentes parámetros, hasta que se encontró un modelo aceptable, el cual se documenta a continuación:

Parámetros: (Ver Anexo F)

La tabla de parámetros de este modelo se refleja en el Anexo F ya que se contiene muchos datos.

Scoring Deciance:



Gráfica 5. Desvianza Modelo NOCLIENT

OUTPUT - TRAINING_METRICS

model	NOCLIENTES_LAPLACE
model_checksum	4087968565457349120
frame	DATA_VAR_NOCLIENTES.
frame_checksum	-238250094392539552
description	·
model_category	Regression
scoring_time	1522998887262
predictions	·
MSE	36590754.861990
RMSE	6049.029250
nobs	1083023
custom_metric_name	·
custom_metric_value	0
r2	0.020082
mean_residual_deviance	1606.289369
mae	1606.289369
rmsle	NaN

OUTPUT - VALIDATION_METRICS

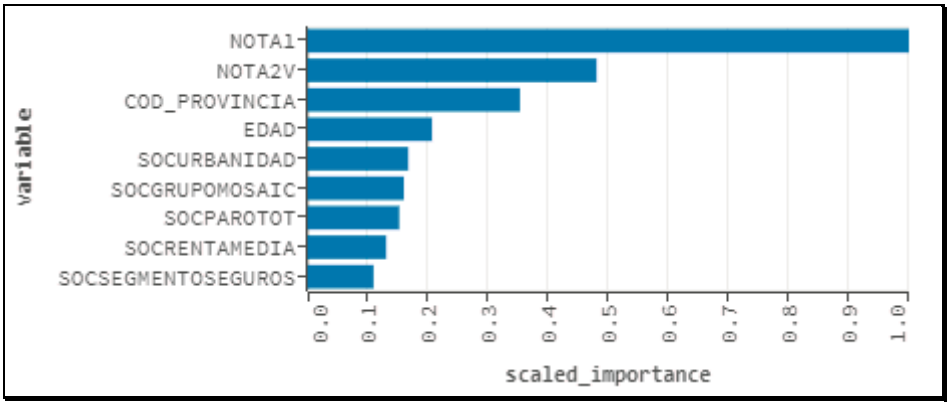
model	NOCLIENTES_LAPLACE
model_checksum	4087968565457349120
frame	DATA_VAR_NOCLIENTES.
frame_checksum	4265599239078777856
description	·
model_category	Regression
scoring_time	1522998887288
predictions	·
MSE	40733648.455889
RMSE	6382.291787
nobs	180477
custom_metric_name	·
custom_metric_value	0
r2	0.014331
mean_residual_deviance	1664.668678
mae	1664.668678
rmsle	NaN

Tabla 14. Métricas fases Entrenamiento y Validación Modelo NOCLIENT

Variables de importancia:

variable	relative_importance	scaled_importance	percentage
NOTA1	114638.4375	1.0	0.3622
NOTA2V	55093.7031	0.4806	0.1741
COD_PROVINCIA	40485.7500	0.3532	0.1279
EDAD	23719.3027	0.2069	0.0749
SOCURBANIDAD	19191.3301	0.1674	0.0606
SOCGRUPOMOSAIC	18349.8359	0.1601	0.0580
SOCPAROTOT	17467.8945	0.1524	0.0552
SOCRENTAMEDIA	14982.2334	0.1307	0.0473
SOCSEGMENTOSEGUROS	12566.4072	0.1096	0.0397

Tabla 15. Variables de importancia Modelo NOCLIENT



Gráfica 6. Gráfica Variables de importancia Modelo NOCLIENT

Predicciones y Evaluación:

Ejecutadas las predicciones, en la siguiente tabla mostramos

Prediction

Actions: Inspect

PREDICTION

model	NOCLIENTES_LAPLACE
model_checksum	4087968565457349120
frame	DATA_VAR_NOCLIENTES.hex
frame_checksum	7980856898598173696
description	-
model_category	Regression
scoring_time	1523040235006
predictions	prediction-c6671bc0-7279-4161-a4c9-9048cda7d354
MSE	36984129.954795
RMSE	6081.457881
nobs	1804792
custom_metric_name	-
custom_metric_value	0
r2	0.018214
mean_residual_deviance	1621.867610
mae	1621.867610
rmsle	NaN

Tabla 16. Resultado Predicciones Modelo NOCLIENT.

Desviación en el acierto en las predicciones.

Comparamos la predicción con el valor real y generamos 5 tramos de variación, con los resultados que se observan en la siguiente tabla:

% Diferencia	<1%	1-5%	5-10%	10-15%	>15%	Total
Total Observaciones	1.509.103	215.026	41.092	14.201	25.370	1.804.792
% del Total de Observaciones	83,62%	11,91%	2,28%	0,79%	1,41%	100,00%

Tabla 17 Desviación predicciones Modelo NOCLIENT

En la anterior tabla observamos que en el 83,62% del total de observaciones, la variación entre el dato real y la predicción es inferior al 1%, y para un 11,91% la variación es menor al 5% por tanto podemos decir que el nivel de acierto del modelo es muy bueno.

6.4.1.3 Modelo 3 POSITIVE

Cuando la persona se ha identificado como cliente de Reale y el algoritmo de probabilidad ha retornado 1, se ejecuta este algoritmo que va a predecir cuál es el margen positivo que va a dejar el cliente.

Al igual que los anteriores algoritmos, aquí también hemos utilizado Auto Machine Learning de H2o para generar diversos modelos y finalmente modificar sus parámetros, incluyendo la función de distribución “gamma” con la cual se han observado el mejor nivel de predicción.

A continuación, solo documentamos el modelo final.

Parámetros:

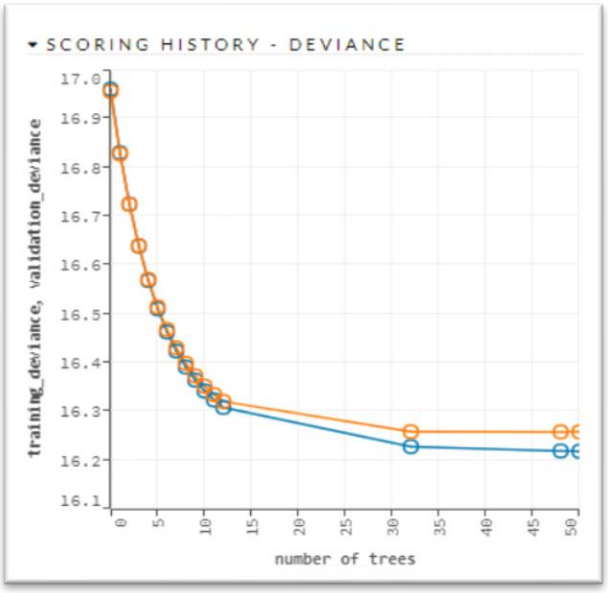
▾ MODEL PARAMETERS

Show all parameters

Parameter	Value	Description
model_id	gbm-4be2eac4-eb10-4c5f-ba17-a95b36c883ef	Destination id for this model; auto-generated if not specified.
nfolds	5	Number of folds for K-fold cross-validation (0 to disable or >= 2).
response_column	MARGEN_BRUTO	Response variable column.
max_depth	16	Maximum tree depth.
min_rows	100	Fewest allowed (weighted) observations in a leaf.
seed	-7213458750158757000	Seed for pseudo random number generator (if applicable)
distribution	gamma	Distribution function
sample_rate	0.8	Row sample rate per tree (from 0.0 to 1.0)
col_sample_rate	0.7	Column sample rate (from 0.0 to 1.0)
custom_metric_func		Reference to custom evaluation function, format: `language:keyName=funcName`

Tabla 18 Parámetros Modelo POSITIVE

Scoring Deciance:



Gráfica 7. Scoring Modelo POSITIVE

OUTPUT - TRAINING_METRICS

model	gbm-4be2eac4-eb10-4c5f-ba17-a95b36c883ef
model_checksum	-2374945623369162752
frame	·
frame_checksum	0
description	·
model_category	Regression
scoring_time	1522837226905
predictions	·
MSE	1659818.572690
RMSE	1288.339463
nobs	1070501
custom_metric_name	·
custom_metric_value	0
r2	0.647163
mean_residual_deviance	16.217952
mae	690.936108
rmsle	0.820221

OUTPUT - VALIDATION_METRICS

model	gbm-4be2eac4-eb10-4c5f-ba17-a95b36c883ef
model_checksum	-2374945623369162752
frame	·
frame_checksum	0
description	·
model_category	Regression
scoring_time	1522837226961
predictions	·
MSE	1796262.730603
RMSE	1340.247265
nobs	305318
custom_metric_name	·
custom_metric_value	0
r2	0.597346
mean_residual_deviance	16.257944
mae	733.476396
rmsle	0.849038

Tabla 19. Métricas fases Entrenamiento y Validación Modelo POSITIVE

Métricas fase de Cross-Validation:

OUTPUT - CROSS-VALIDATION METRICS SUMMARY

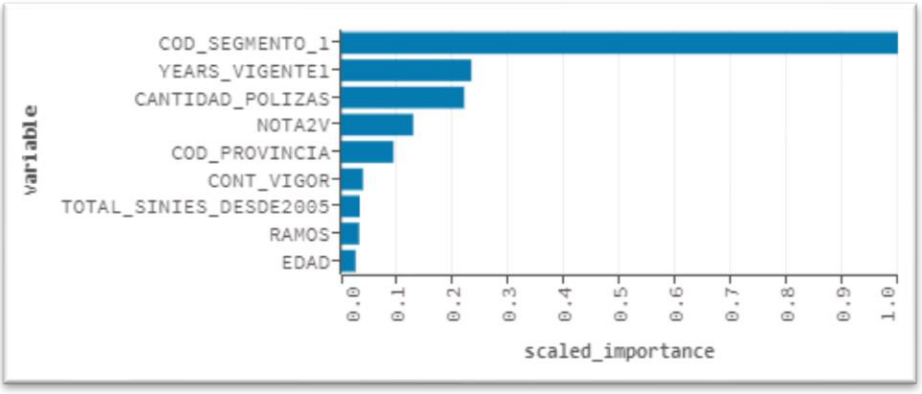
	mean	sd	cv_1_valid	cv_2_valid	cv_3_valid	cv_4_valid	cv_5_valid
mae	738.79364	2.1564496	741.81903	738.2103	734.98517	736.1927	742.76105
mean_residual_deviance	16.26223	0.0019586184	16.264442	16.262665	16.258455	16.259758	16.265825
mse	1944110.1	82578.875	1892436.2	2060365.5	1821467.9	1840323.8	2105957.2
r2	0.5870658	0.008216922	0.5968051	0.57749426	0.5988656	0.5928689	0.5692951
residual_deviance	16.26223	0.0019586184	16.264442	16.262665	16.258455	16.259758	16.265825
rmse	1393.6901	29.479149	1375.6584	1435.3973	1349.6177	1356.5853	1451.1917
rmsle	0.8480865	0.0011803035	0.85117424	0.847443	0.84821194	0.8473783	0.8462249

Tabla 20. Métricas fase Cross Validation Modelo POSITIVE

Variables de Importancia:

variable	relative_importance	scaled_importance	percentage
COD_SEGMENTO_1	2118368.7500	1.0	0.5528
YEARS_VIGENTE1	495563.8750	0.2339	0.1293
CANTIDAD_POLIZAS	468169.4375	0.2210	0.1222
NOTA2V	273863.3750	0.1293	0.0715
COD_PROVINCIA	198839.6250	0.0939	0.0519
CONT_VIGOR	82785.5625	0.0391	0.0216
TOTAL_SINIES_DESDE2005	70939.5000	0.0335	0.0185
RAMOS	68393.0547	0.0323	0.0178
EDAD	54996.5078	0.0260	0.0144

Tabla 21. Variables de importancia Modelo POSITIVE



Gráfica 8. Variables de importancia Modelo POSITIVE

Ejemplos de los árboles generados:

Mostramos ahora algunos ejemplos de los árboles generados con este modelo:

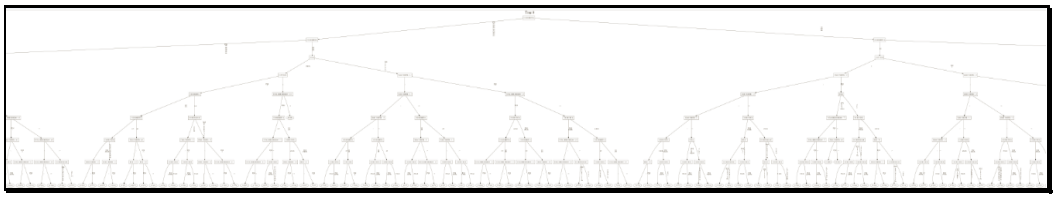


Figura 34. Ejemplo 1 Árbol generado modelo POSITIVE

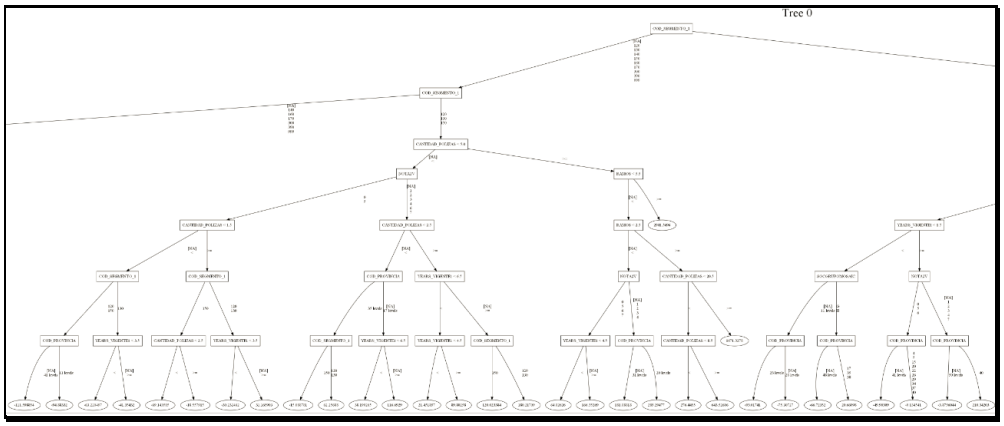


Figura 35. Ejemplo 2 Árbol generado modelo POSITIVE

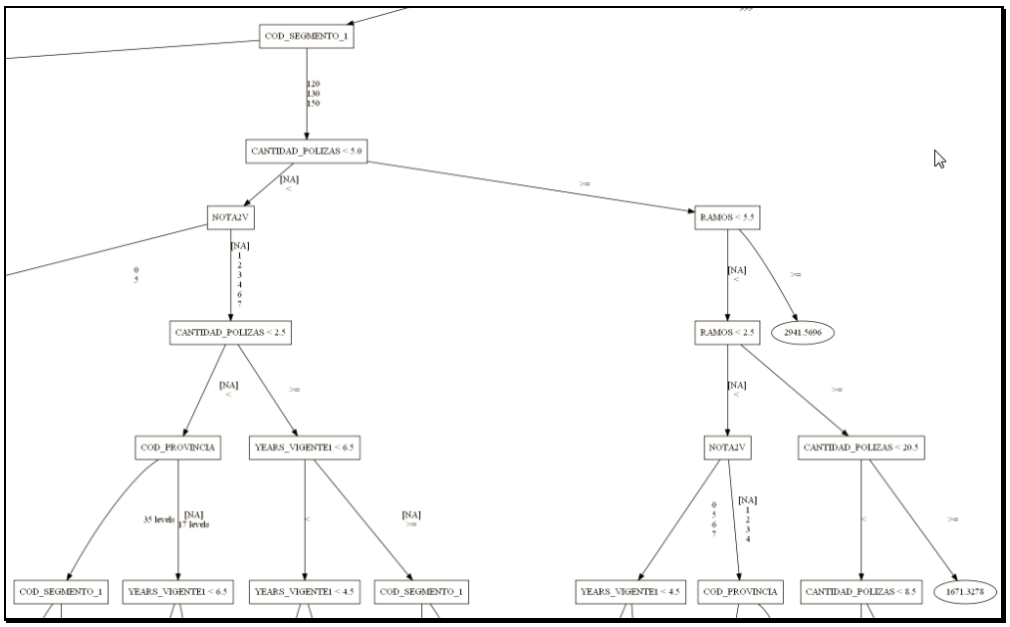


Figura 36. Ejemplo 3 Árbol generado modelo POSITIVE

Evaluación:

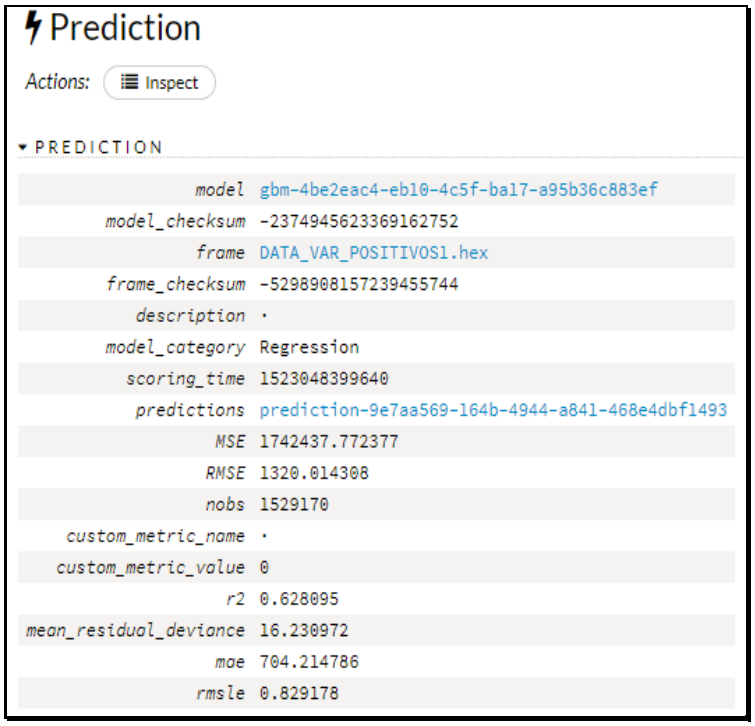


Figura 37. Predicciones Modelo POSITIVE

En la siguiente tabla presentamos el análisis de la predicción en cuanto a porcentajes de desviación con respecto del Margen Bruto el real.

% Diferencia	<1%	1-5%	5-10%	10-15%	>15%	Total
Total Observaciones	1203133	236786	42033	16139	31079	1.529.170
% del Total de Observaciones	78,68%	15,48%	2,75%	1,06%	2,03%	100,00%

Tabla 22. Desviación predicciones Modelo POSITVE

Como podemos observar la predicción para un 94,16% varía en menos del 5% y estos resultados se consideran buenos, para el propósito de este proyecto.

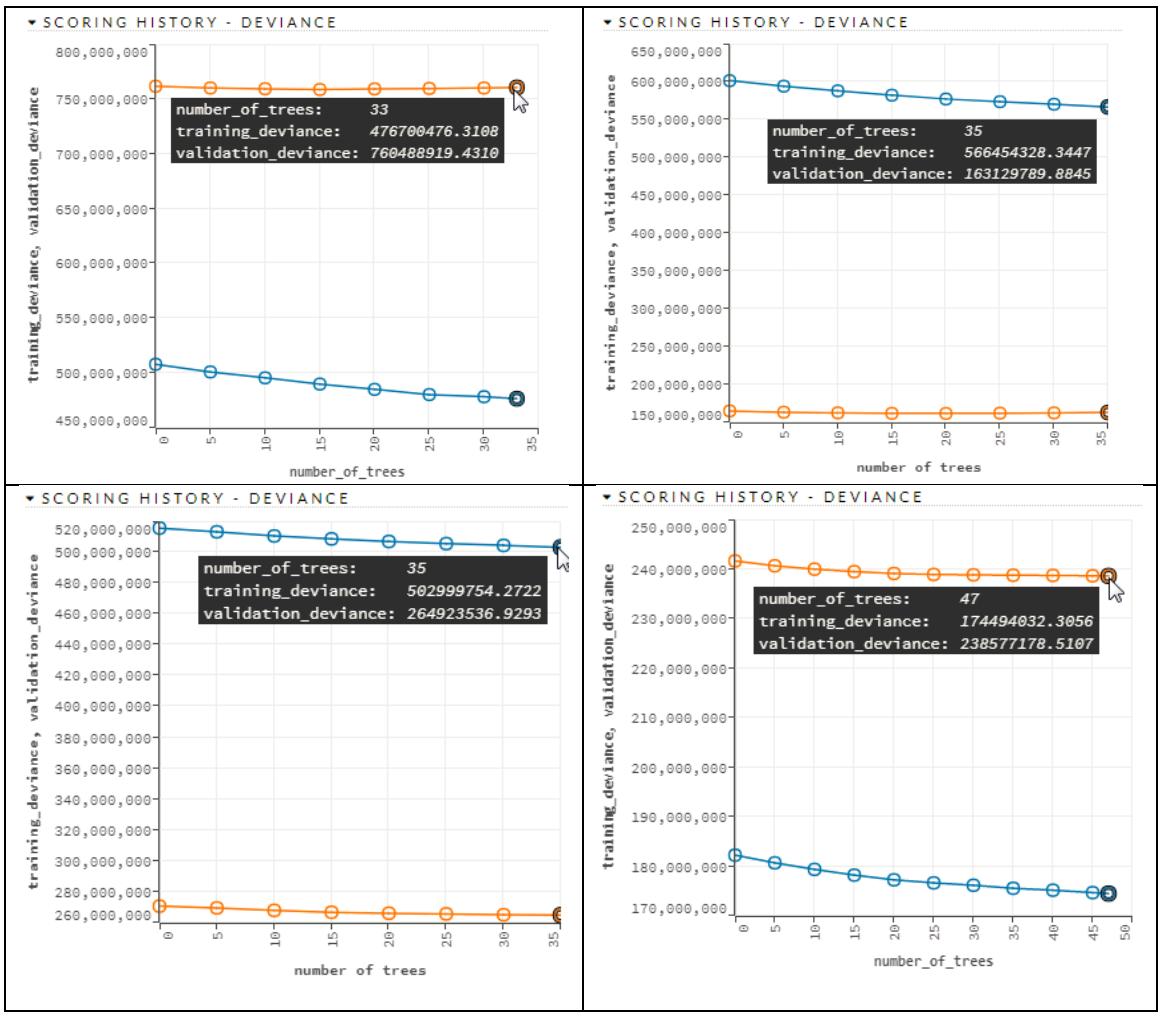
6.4.1.4 Modelo 4 NEGATIVE

Este modelo se ejecuta cuando se ha identificado que es o ha sido cliente de Reale y la Probabilidad de Margen es Negativo; aunque la probabilidad del margen es negativo, este modelo es capaz de predecir tanto negativos como positivos, según hemos observado en los cientos de ejecuciones y comprobaciones que se han realizado.

Esta ha sido el modelo más complicado de realizar, primero porque se cuenta con un dataset muy pequeño para entrenar el modelo y adicionalmente por los outlier presentes en los datos

existentes; con estas características fue necesario, crear varios datasets, en los cuales se suavizaba el efecto de los outlier, eliminando los registros de los extremos.

Para que se aprecie esta complejidad a continuación mostramos una tabla con graficas de cuatros de los modelos probados, donde se puede observar una alta desviación entre los modelos de training y validación en la escala del algoritmo de verosimilitud.

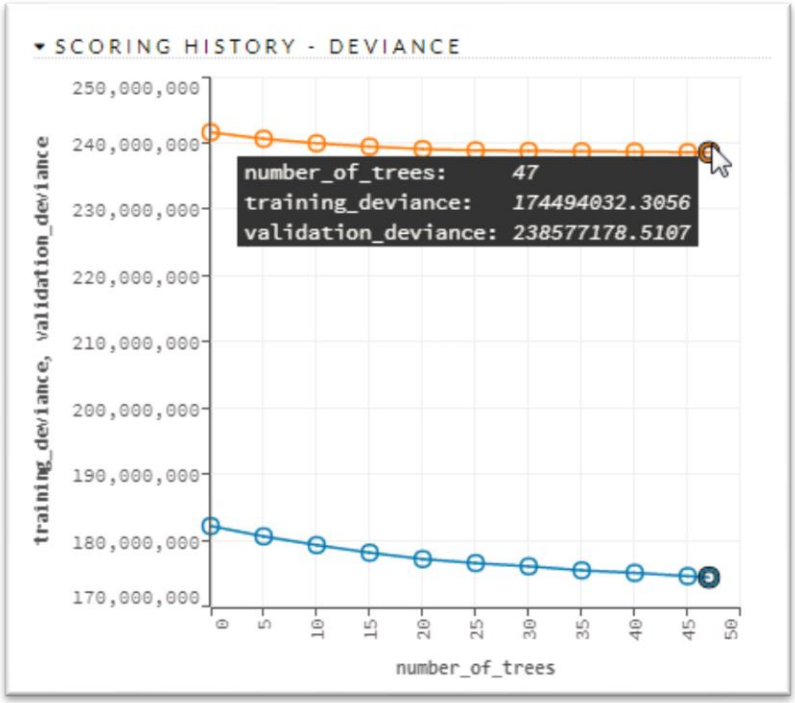


A continuación documentamos el modelo el modelo inicialmente implementado, que es el que menor desviación presentaba en entre el training y la validación.

Parámetros:

Parameter	Value	Description
model_id	GBM_grid_0_AutoML_20180407_123326_model_8	Destination id for this model; auto-generated if not specified.
training_frame	automl_training_RTMP_sid_b24e_13	Id of the training data frame.
validation_frame	automl_validation_RTMP_sid_b24e_13	Id of the validation data frame.
nfolds	5	Number of folds for K-fold cross-validation (0 to disable or >= 2).
keep_cross_validation_predictions	true	Whether to keep the predictions of the cross-validation models.
score_tree_interval	5	Score the model after every so many trees. Disabled if set to 0.
fold_assignment	Modulo	Cross-validation fold assignment scheme, if fold_column is not specified. The 'Stratified' option will stratify the folds based on the response variable, for classification problems.
response_column	MARGEN_BRUTO	Response variable column.
ntrees	47	Number of trees.
max_depth	7	Maximum tree depth.
min_rows	100	Fewest allowed (weighted) observations in a leaf.
stopping_metric	MSE	Metric to use for early stopping (AUTO: logloss for classification, deviance for regression)
stopping_tolerance	0.0027506179269605616	Relative tolerance for metric-based stopping criterion (stop if relative improvement is not at least this much)
seed	789101234	Seed for pseudo random number generator (if applicable)
learn_rate	0.05	Learning rate (from 0.0 to 1.0)
distribution	gaussian	Distribution function
sample_rate	0.7	Row sample rate per tree (from 0.0 to 1.0)
col_sample_rate	0.4	Column sample rate (from 0.0 to 1.0)
col_sample_rate_per_tree	0.7	Column sample rate per tree (from 0.0 to 1.0)
min_split_improvement	0.0001	Minimum relative improvement in squared error reduction for a split to happen

Scoring Deviance:



▾ OUTPUT - TRAINING_METRICS

model	GBM_grid_0_AutoML_20180407_123326_model_8
model_checksum	7801789937194343424
frame	automl_training_RTMP_sid_b24e_13
frame_checksum	4996618214036011008
description	·
model_category	Regression
scoring_time	1523097552843
predictions	·
MSE	174494032.305623
RMSE	13209.618931
nobs	132172
custom_metric_name	·
custom_metric_value	0
r2	0.042538
mean_residual_deviance	174494032.305623
mae	3693.281464
rmsle	NaN

▾ OUTPUT - VALIDATION_METRICS

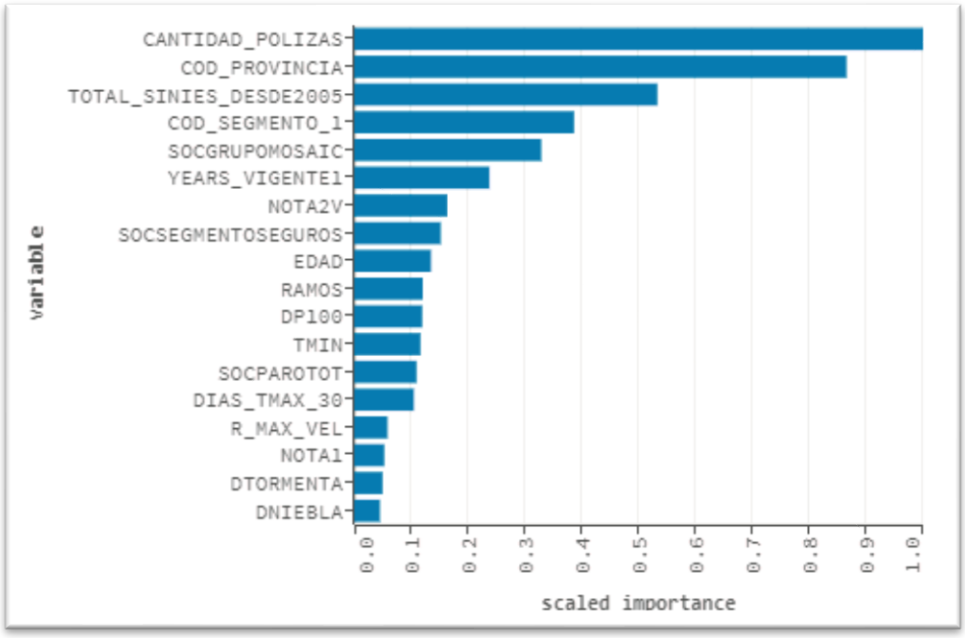
model	GBM_grid_0_AutoML_20180407_123326_model_8
model_checksum	7801789937194343424
frame	automl_validation_RTMP_sid_b24e_13
frame_checksum	-8382978159538304000
description	·
model_category	Regression
scoring_time	1523097552849
predictions	·
MSE	238577178.510727
RMSE	15445.943756
nobs	33210
custom_metric_name	·
custom_metric_value	0
r2	0.012508
mean_residual_deviance	238577178.510727
mae	3787.209665
rmsle	NaN

▾ OUTPUT - CROSS_VALIDATION_METRICS

model	GBM_grid_0_AutoML_20180407_123326_model_8
model_checksum	7801789937194343424
frame	automl_training_RTMP_sid_b24e_13
frame_checksum	4996618214036011008
description	5-fold cross-validation on training data
model_category	Regression
scoring_time	1523097552852
predictions	·
MSE	178260066.820340
RMSE	13351.406923
nobs	132172
custom_metric_name	·
custom_metric_value	0
r2	0.021874
mean_residual_deviance	178260066.820340
mae	3725.684228
rmsle	NaN

Variables de Importancia:

variable	*relative_importance*	*scaled_importance*	*percentage*
CANTIDAD_POLIZAS	2036071137280.0	1.0	0.2186
COD_PROVINCIA	1762082816000.0	0.8654	0.1892
TOTAL_SINIES_DESDE2005	1084549562368.0	0.5327	0.1165
COD_SEGMENTO_1	786084724736.0	0.3861	0.0844
SOCGRUPOMOSAIC	668822929408.0	0.3285	0.0718
YEARS_VIGENTE1	482862366720.0	0.2372	0.0519
NOTA2V	331950718976.0	0.1630	0.0356
SOCSEGMENTOSEGUROS	308988837888.0	0.1518	0.0332
EDAD	274109906944.0	0.1346	0.0294
RAMOS	244393476096.0	0.1200	0.0262
DP100	242592120832.0	0.1191	0.0261
TMIN	236303007744.0	0.1161	0.0254
SOCPAROTOT	222356488192.0	0.1092	0.0239
DIAS_TMAX_30	212892712960.0	0.1046	0.0229
R_MAX_VEL	118574874624.0	0.0582	0.0127
NOTA1	107096883200.0	0.0526	0.0115
DTORMENTA	100766269440.0	0.0495	0.0108
DNIEBLA	91538857984.0	0.0450	0.0098



Evaluación:

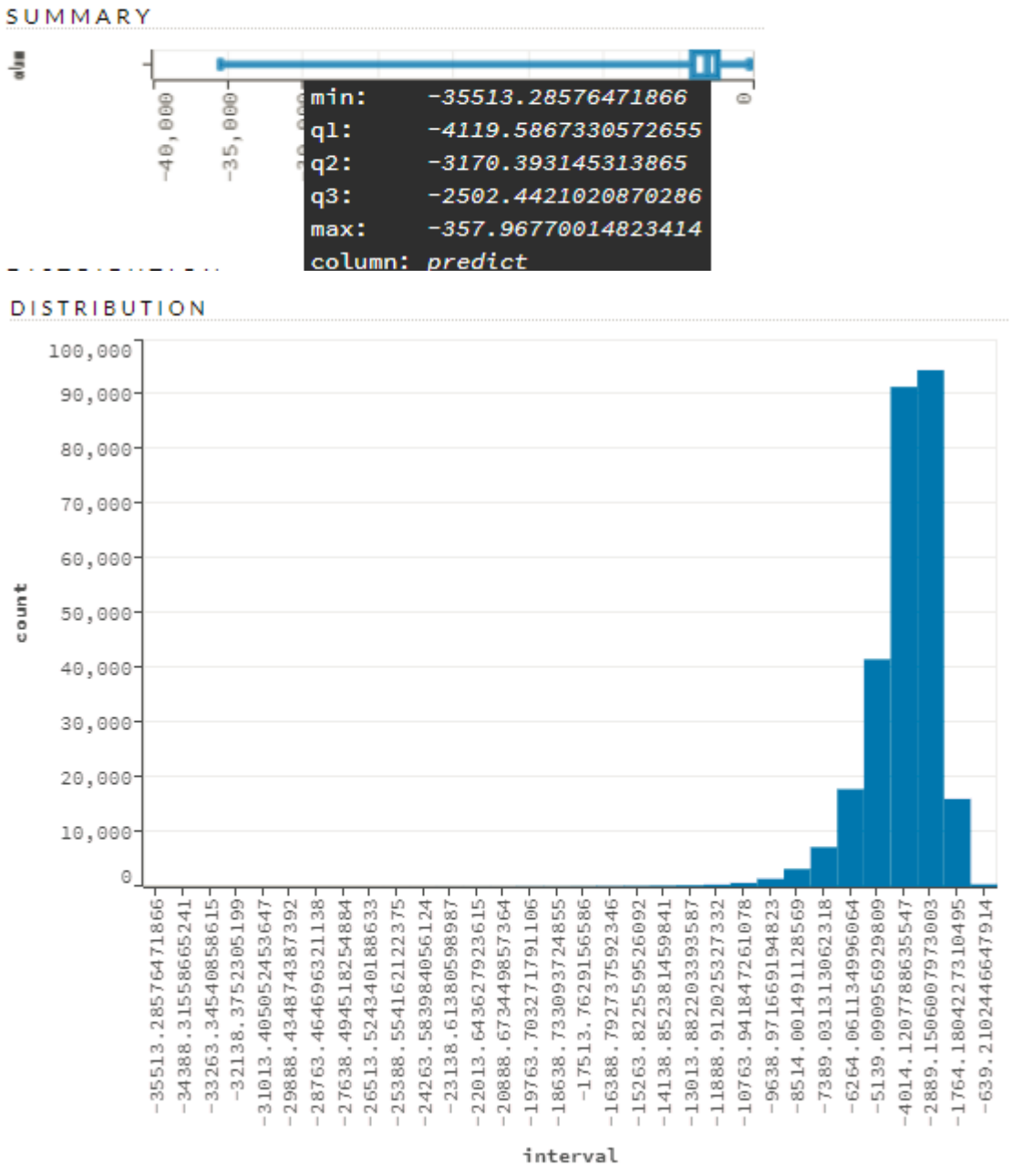
Aplicamos Predicciones a todo el dataset y obtenemos los siguientes resultados:

▾ PREDICTION

model	GBM_grid_0_AutoML_20180407_123326_model_8
model_checksum	7801789937194343424
frame	DATA_VAR_NEGATIVOS_sid_b24e_2
frame_checksum	2943520362885767680
description	·
model_category	Regression
scoring_time	1523103162669
predictions	prediction-f66b78cc-97b9-446a-9830-390c8396a271
MSE	191701778.929799
RMSE	13845.641153
nobs	275622
custom_metric_name	·
custom_metric_value	0
r2	0.027450
mean_residual_deviance	191701778.929799
mae	3720.698047
rmsle	NaN

▾ COLUMN SUMMARIES

label	*type*	*Missing*	*Zeros*	*+Inf*	*-Inf*	*min*	*max*	*mean*	*sigma*	*cardinality*	*Actions*
predict	real	0	0	0	0	-35513.2858	-357.9677	-3495.3909	1686.4382		· ·



Este es el mejor modelo que se ha conseguido, pero sigue sin ser buen modelo, este modelo se tendra que seguir probando con otras técnicas estadisticas, guiado por los expertos en esta área.

6.5 Desarrollo

En esta etapa del proyecto una vez conseguidos los modelos machine learning, se procede a maquetar los diferentes componentes que se reflejaron en el diagrama de flujo del proyecto, Figura 15, y esto lo abordamos con dos desarrollos una página Web y una aplicación móvil que describiremos en los siguientes apartados.

6.5.1 Página web

URL página WEB: https://pilotosmartpricingweb.azurewebsites.net/

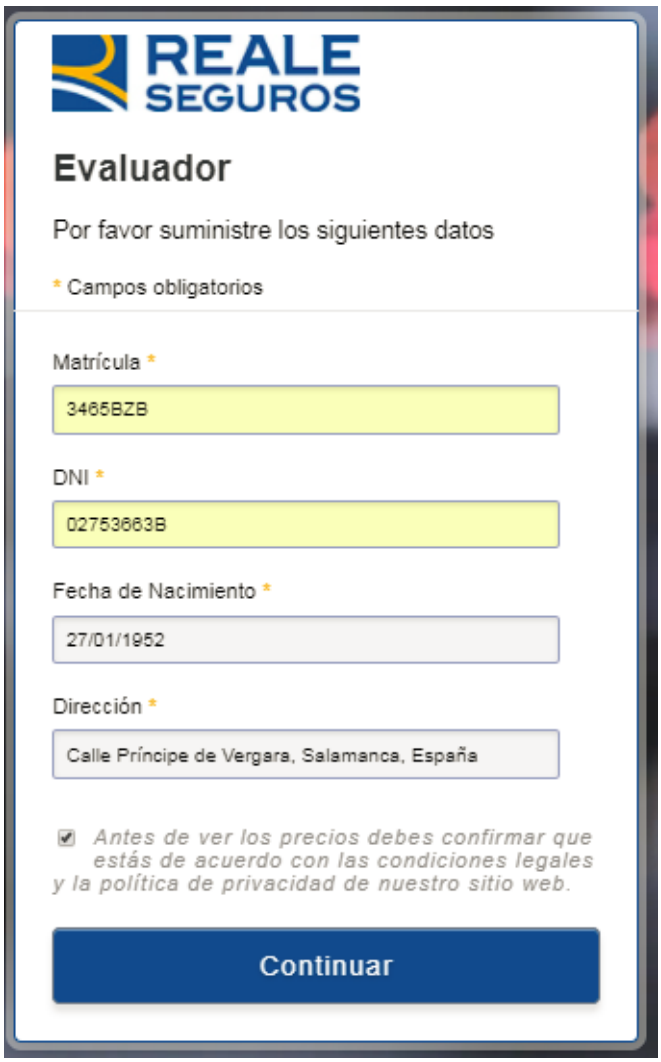


Figura 38. Web entrada Tarificador

Esta página Web ha sido diseñada, pensando en su adaptación a dispositivos móviles, para aquellos que no se puedan instalar la aplicación móvil que se ha desarrollado y se explica a continuación de este apartado.

En esta página se incorporan algunas validaciones de DNI, de matrículas de coches, calendario para selección de fechas e integración con Api de google para autocompletado en la dirección.

Una vez introducidos los 4 datos requeridos (matricula, DNI, Fecha de Nacimiento y Dirección) y aceptadas las políticas de privacidad, empieza la ejecución del flujo del proyecto. Que consistiría principalmente en:

- Invocar Webservice del centro de Zaragoza para recuperar las características del coche que requiere el tarificador.
- Invocar Webservice de Experian para rescatar información de morosidad
- Consultar base de datos sociodemograficos
- Consultar base de datos metereologicos
- Invocar Webservice de Catastro
- Invocar al tarificador de Reale

En cada uno de los anteriores Webservices se van recopilando la información de las variables requeridas para los algoritmos y en general para la ejecución completa del flujo hasta retornar una cotización.

Llegados a este punto, a efectos de este trabajo de fin de master, por diversos motivos ajenos a los autores de este documento, no se puede tener acceso a esos Webservices que obtienen la información requerida para los algoritmos, hemos subsanado esta carencia con una colección Cosmos DB, donde tenemos un amplio set de datos para pruebas con todas las variables requeridas para hacer pruebas, esta colección llamada “variables_modelos_ml”, se utiliza tanto en la página web como en la App móvil; por tanto los datos retornados siempre son de Test.

En la siguiente gráfica se puede observar, la recuperación de las variables y la invocación a los algoritmos de machine learning.

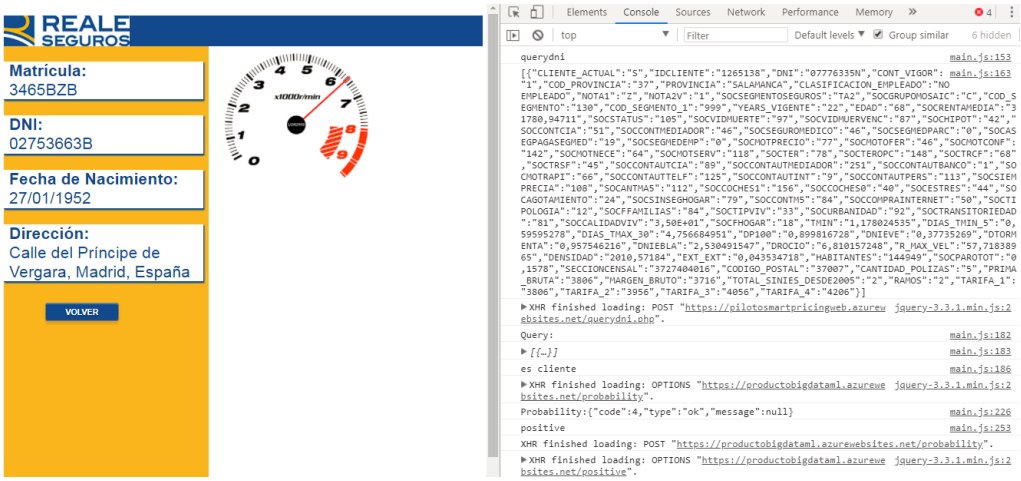


Figura 39. Web, espera cálculo del seguro.

En la siguiente imagen se puede observar mejor la recuperación de los datos de la base de datos de pruebas y las invocaciones a las API de los modelos de Machine Learning, desplegadas en Azure.

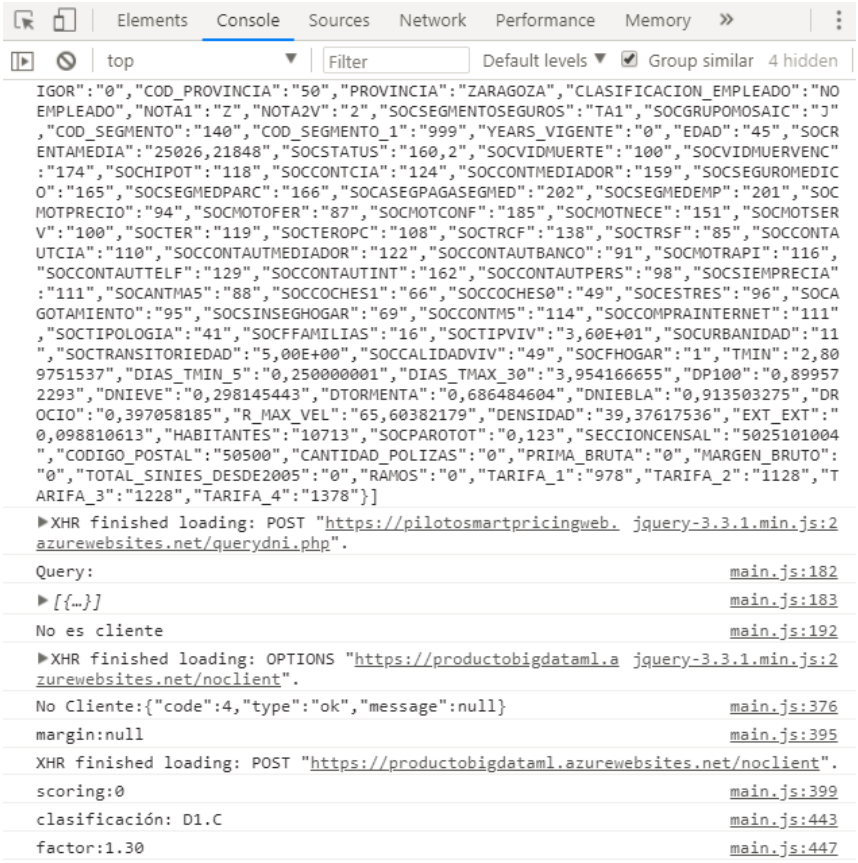


Figura 40 Recuperación de datos

El resultado final es un presupuesto con 4 opciones de modalidad de contratación.

Figura 41. Web. Resultado presupuesto de las distintas coberturas

El Resultado final debe un precio óptimo para el cliente y para Reale, por tanto si el algoritmo determina que es un cliente que su margen no será positivo, le va a penalizar la tarifa, como ha sido el caso del ejemplo adjunto a este documento.

6.5.2 Aplicación móvil

En este apartado vamos a entrar en detalle sobre el funcionamiento de la app móvil. Hablaremos además de aquellos procesos que se realizan para el funcionamiento. Aquí se sigue el flujo que se ha explicado en figura 15 de la memoria.

Figura 42. App Móvil

Comenzamos con la generación de una app móvil utilizando la herramienta Android Studio para el diseño de la interfaz. Se trata de generar algo sencillo y fácil de utilizar, que nos permita operar con el menor detalle de datos personales posible que es uno de los objetivos marcados para este proyecto.

Se genera una primera pantalla que como vemos en el flujo anterior, tendrá como input los siguientes datos:

- Nº Matricula
- DNI
- Fecha Nacimiento
- Dirección

Dado que se debe cumplir con la LOPD (Ley Orgánica de Protección de Datos), se mostrará un check que el usuario debe marcar para poder continuar con la solicitud, ya que es necesario tener consentimiento por parte del usuario para manipular sus datos personales.

Una vez introducidos todos los datos, empieza la ejecución del flujo del proyecto. Que consistiría, al igual que en ocurre en la web en:

- Invocar Webservice del centro de Zaragoza para recuperar las características del coche que requiere el tarificador.
- Invocar Webservice de Experian para rescatar información de morosidad
- Consultar base de datos sociodemograficos
- Consultar base de datos metereologicos
- Invocar Webservice de Catastro
- Invocar al tarificador de Reale

En cada uno de los anteriores Webservices se van recopilando la información de las variables requeridas para los algoritmos y en general para la ejecución completa del flujo hasta retornar una cotización.

Llegados a este punto, a efectos de este trabajo de fin de master, por diversos motivos ajenos a los autores de este documento, no se puede tener acceso a esos Webservices que obtienen la información requerida para los algoritmos, hemos subsanado esta carencia con una colección Cosmos DB, donde tenemos un amplio set de datos para pruebas con todas las variables requeridas para hacer pruebas, esta colección llamada “variables_modelos_ml”, se utiliza tanto en la página web como en esta App móvil; por tanto los datos retornados siempre son de Test.

Con los datos facilitados, realizaremos diferentes acciones que iremos detallando a continuación. El fin de este flujo es retornar una simulación que será optimizada con respecto a los precios que hoy en día ofrece el tarificador de Reale. Optimizado no significa “más barato” ya que en todo momento hablamos de mejorar el margen que genera el cliente dentro de la empresa.

Podemos encontrarnos un cliente que por falta de siniestros y partes generados obtenga una prima que rebaje la simulación de su precio actual, así como un cliente que por su reiteración en siniestros y números de partes generados a la empresa obtenga una prima que aumente la simulación de su actual precio.

Una vez terminados los procesos intermedios y calculado el nuevo Scoring que se aplicara al cliente, generaremos una pantalla fin con los siguientes datos de salida:

- Terceros Básica
- Terceros con Lunas
- Terceros Ampliados
- Todo riesgo con Franquicia

A continuación, se muestra en la figura 34 una imagen de las colecciones que se han generado y de las que hablaremos un poco más en detalle.

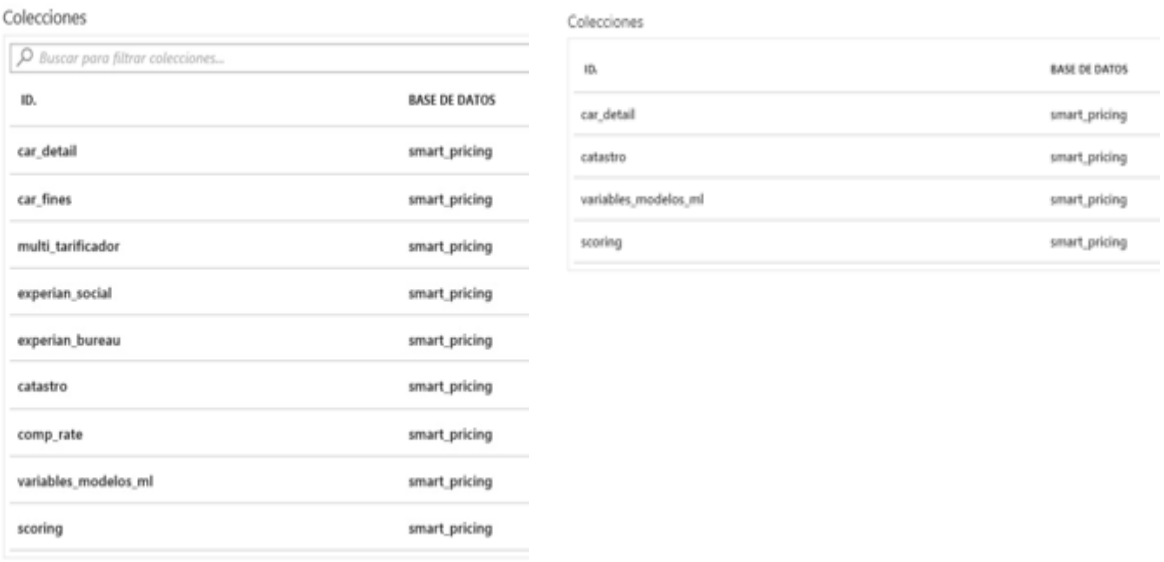


Figura 43. Colecciones CosmoDB

En el proyecto original se iban a utilizar todas y cada una de las colecciones, pero finalmente por falta de acuerdo a la hora de abordar el piloto, utilizaremos solo cuatro de ellas, al menos inicialmente.

Colección CosmoDB CATASTRO

Se genera una colección llamada catastro que albergara los datos recuperados del Webservice oficial de dicha página. Este servicio web es un servicio gratuito opensource, cuyos datos serán utilizados para calcular los datos de seguros de hogar. Aunque inicialmente este proyecto no tiene como fin este sector, al ser un objetivo futuro para este proyecto se aborda dicha funcionalidad.

El proceso para consumir el Webservice se realiza en código Python. Los parámetros utilizados para invocar este servicio y los parámetros de salida vienen especificados en la página web oficial del lugar. Se pueden ver en el Anexo B del documento.

La información que devuelve está generada en formato XML. Al estar utilizando CosmosDB necesitamos un documento en formato JSON por lo que este proceso deberá convertir dicho documento para poder almacenarlo.

```
jsonString = json.dumps(xmltodict.parse(resp.text))
```

Se validará que los datos existan. Si existen, modificará los datos con los recuperados de la nueva consulta. En caso de no existir, insertará una nueva línea en la colección. De esta forma siempre tendremos los datos actualizados para su utilización.

Se muestra una imagen sobre los documentos almacenados en dicha colección:

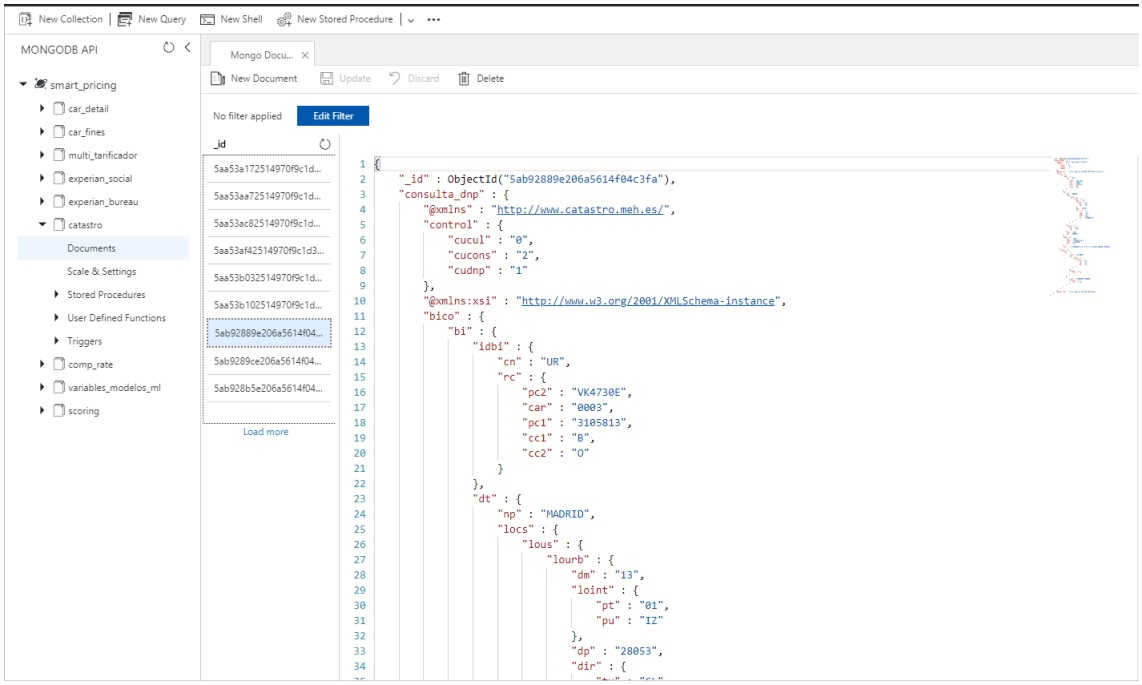


Figura 44.Detalle datos colección catastro de CosmosDB

A continuación, en la siguiente figura, mostramos el código Python con el que realizamos la consulta y generación de los datos en la colección. Como se puede comprobar, mantenemos la línea de los datos de entrada mencionados anteriormente.

```
@app.route('/catastro' , methods=['GET'])
def catastro():
        provincia = request.args.get('provincia')
        municipio = request.args.get('municipio')
        via = request.args.get('via')
        calle = request.args.get('calle')
        numero = request.args.get('numero')
        bloque = request.args.get('bloque')
        escalera = request.args.get('escalera')
        piso = request.args.get('piso')
        puerta = request.args.get('puerta')
        uri = "mongodb://rsgdsmongodb:gw1kdadbuISj2TjK0Wt6nKTnuuPk4w8EWPDwU0ixtQqM3c2d8DzGCbxIjo7Aw1V7jRknPn
        client = MongoClient(uri)
        db = client.smart_pricing
        col = db.catastro
        url = 'http://ovc.catastro.meh.es/ovcservweb/ovcswlocalizacionrc/ovccallejero.asmx/Consulta_DNPLOC'
        head = {"Content-type": "application/xml"}
        parameters = {'Provincia':provincia,
                      'Municipio':municipio,
                      'Sigla':via,
                      'Calle':calle,
                      'Numero':numero,
                      'Bloque':bloque,
                      'Escalera':escalera,
                      'Planta':piso,
                      'Puerta':puerta}
        resp = requests.get(url,params=parameters,headers=head)
        jsonString = xmltodict.parse(resp.text)
        ldt = jsonString['consulta_dnp']['bico']['bi']['ldt']
        jsonString = json.dumps(xmltodict.parse(resp.text))
        _id = col.find_one({"consulta_dnp.bico.bi.ldt" : ldt})
        if _id is None:
            col.insert_one(json.loads(jsonString))
        else:
            col.update_one({"_id":_id},{"$set":json.loads(jsonString)},upsert=False)
        client.close()
        return json.dumps(jsonString)
```

Figura 45. Código Python consulta y generación de datos de la colección catastro de MongoDB.

El return contendrá el objeto json que se insertará en la colección cosmosDB.

Colección CosmosDB VARIABLES_MODELOS_ML

Esta colección contiene todas y cada una de las variables susceptibles de ser utilizadas para la generación de los algoritmos que posteriormente nos darán la optimización de la simulación.

Como inicialmente en la fase de construcción de los modelos, no conocíamos las variables definitivas que tendríamos que utilizar, por lo que optamos por realizar la recuperación de todas las variables que contiene la colección.

Aunque finalmente y una vez claras dichas variables, podríamos haber modificado este proceso dejando solo aquellas que realmente invocan los algoritmos, pensamos que dejarlas en su totalidad podría servir para modificaciones futuras o nuevos algoritmos a incluir que necesitaran de esta información, lo cual no sería un problema y tampoco tendríamos que modificar esta función.

Mostramos la siguiente figura con los datos de esta colección:

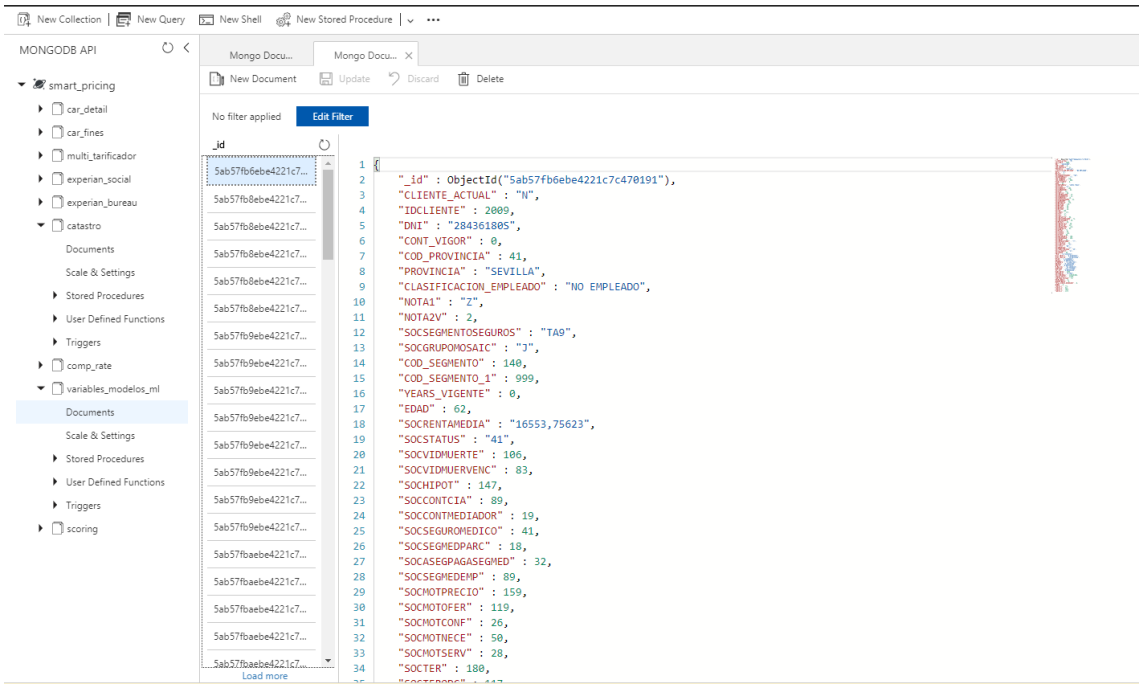


Figura 46. Colección Variables_modelos_ML de CosmosDB.

Mediante código Python recuperamos las variables y las almacenamos en la función bigdatahtml que explicaremos posteriormente. Este código se refleja en el Anexo C del documento.

En esta función, si el dato no existe, para hacer funcionar el piloto, se recuperará de la primera fila encontrada y se proporcionará siempre una simulación. Los datos incluidos en la BBDD son de prueba, hay que tener en cuenta esta casuística que puede darse sin duda.

Una vez almacenadas las variables ya podemos operar con ellas y aplicar la lógica para la ejecución de unos algoritmos u otros. Esta ejecución será en función, principalmente, de si el cliente, es decir, pertenece a Reale o no.

Como el propio sentido común nos dice, ejecutaremos un mayor número de algoritmos si el cliente es un Cliente nuestro, ya que deberemos recuperar toda aquella información que ya tenemos almacenada en nuestra bbdd para optar a la optimización de la prima.

A continuación, la funcionalidad del bigdataml que comentábamos anteriormente y que contiene toda la lógica a aplicar se puede ver en el Anexo D de la memoria del proyecto.

Al final de la función el return va a retornar un formato Json con los datos que almacenaremos en la CosmosDB Scoring. Esta colección se utiliza para guardar aquellos datos con los que posteriormente realizaremos las visualizaciones.

Colección CosmoDB SCORING o ARBOL DE DECISION SCORING

Esta colección alberga la información generada por la ejecución de los algoritmos. Calcula el Scoring que se aplicara posteriormente a las tarifas originales generadas por el tarificador de Reale. A partir del factor que nos devuelve el algoritmo y la tarifa original calculamos ese Scoring.

Los rangos de los factores se incluyen en la documentación para información del usuario. Veremos posteriormente la lógica para aplicarlos en el código Python.

```
#la tabla para calcular el factor:
#C1.A = 0.75
#C1.B = 0.85
#C1.C = 1.10
#C2.A = 1.20
#C2.B = 1.35
#C2.C = 1.5
#D1.A = 0.70
#D1.B = 0.85
#D1.C = 1.30
```

Tabla 23. Tabla factores para Scoring

Incluimos, además, datos como Fecha de nacimiento, código postal, etc. Valores que utilizaremos como información para la visualización por medio de Power BI y para realizar la demostración sobre los KPI.

El Json que genera el proceso Scoring tiene este aspecto:

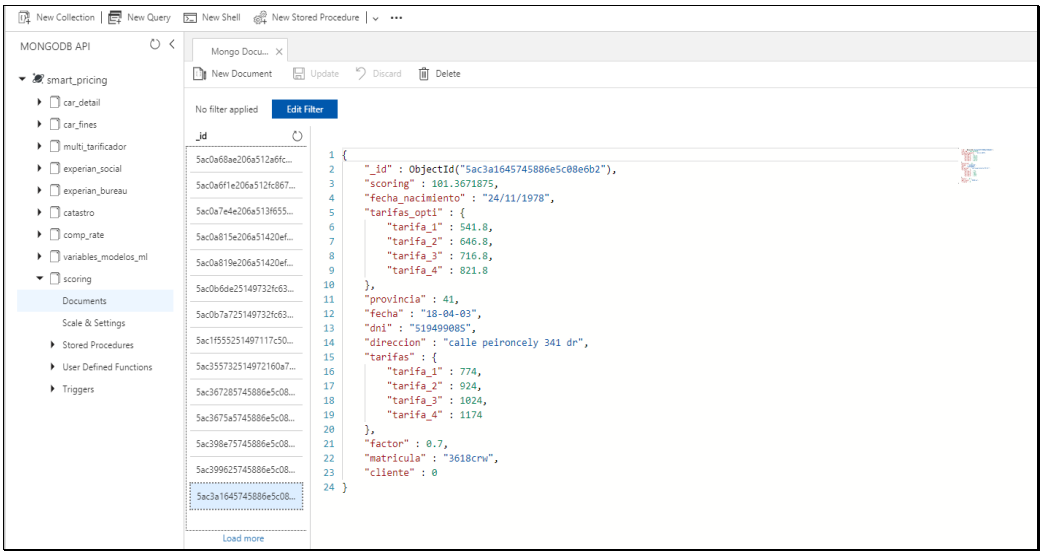


Figura 47. Objeto Json generado por el proceso de Scoring

Por otro lado, el código Python para la generación del proceso del Scoring podemos verlo en la figura 10 del documento.

```
def scoring(resultado,
            tarifa_1,
            tarifa_2,
            tarifa_3,
            tarifa_4,
            aux_cliente,
            dni,
            matricula,
            direccion,
            fecha_nacimiento,
            cod_provincia,
            provincia,
            cliente_actual,
            edad,
            cantidad_polizas,
            total_sinies_desde2005,
            cont_vigor):
        aux_scoring = (float(resultado) / tarifa_3) * 100

        if aux_cliente == 1 and aux_scoring >= 0:
            if aux_scoring >=60:
                aux_factor = 0.75
            if aux_scoring < 60 and aux_scoring >= 25:
                aux_factor = 0.85
            if aux_scoring < 25:
                aux_factor = 1.10
        if aux_cliente == 1 and aux_scoring < 0:
            if aux_scoring >=-36:
                aux_factor = 1.20
            if aux_scoring < -36 and aux_scoring >= -60:
                aux_factor = 1.35
            if aux_scoring < -60:
                aux_factor = 1.5
        if aux_cliente == 0:
            if aux_scoring >=50:
                aux_factor = 0.70
            if aux_scoring < 50 and aux_scoring >= 15:
                aux_factor = 0.85
            if aux_scoring < 15:
                aux_factor = 1.30
```

```
        tarifa_1_opti = tarifa_1 * aux_factor
        tarifa_2_opti = tarifa_2 * aux_factor
        tarifa_3_opti = tarifa_3 * aux_factor
        tarifa_4_opti = tarifa_4 * aux_factor
        pais = 'España'
        uri = "mongodb://rsgdsmongodb:gw1kdadbuISj2TjK0Wt6nKTnuuPk4w8EWPDwU0ixtQqM3c2d8DzGCbxIjo7Aw1V7jRknPn
        client = MongoClient(uri)
        db = client.smart_pricing
        col = db.scoring
        fecha = datetime.datetime.now().strftime("%y-%m-%d")
        jsonString = {
                        "tarifas":{"tarifa_1":tarifa_1,"tarifa_2":tarifa_2,"tarifa_3":tarifa_3,"tarifa_4":tar
                        "tarifas_opti":{"tarifa_1":tarifa_1_opti,"tarifa_2":tarifa_2_opti,"tarifa_3":tarifa_3
                        "scoring":aux_scoring, "factor":aux_factor,"fecha":fecha,"cliente":aux_cliente,
                        "fecha_nacimiento":fecha_nacimiento,
                        "dni":dni,
                        "matricula":matricula,
                        "direccion":direccion,
                        "provincia":cod_provincia,
                        "provincia1":provincia,
                        "cliente_actual":cliente_actual,
                        "edad":edad,
                        "cantidad_polizas":cantidad_polizas,
                        "total_sinies_desde2005":total_sinies_desde2005,
                        "cont_vigor":cont_vigor}

        col.insert_one(json.loads(json.dumps(jsonString)))
        return json.dumps(jsonString)
```

Figura 48.Código Python para el proceso Scoring.

El Json que vemos en la colección CosmosDB es el resultado del return de esta definición. Podríamos añadir tantos datos hiciesen falta para un futuro. Inicialmente con los datos seleccionados creemos que es suficiente para la visualización de estos.

Finalmente vamos a visualizar las llamadas los algoritmos que deciden el factor que aplicaremos a los precios. En total son 4 e iremos viendo cada uno de ellos de forma superficial.

En el apartado xxxxx de la memoria se encuentra la documentación explicativa de estos 4 algoritmos para consultarla.

La función que realiza las llamadas es “bigdatahtml” tal y como explicábamos anteriormente. Para invocar los diferentes algoritmos se utilizarán solo las variables específicas que necesita cada uno de ellos para su funcionamiento.

Identificación de si es cliente

```
def cliente(cliente_actual):
        if cliente_actual == 'N':
            return '0'
        else:
            return '1'
```

Esto significa que cuando el return es un 1, el cliente será de Reale y llamaremos al algoritmo “Probability”.

Cuando el return es un 0, el cliente no es de Reale y llamaremos al algoritmo “NoCliente”. que necesita cada uno de ellos para su funcionamiento.

Algoritmo PROBABILITY

Las variables de entrada para este algoritmo son:

https://productobigdataml.azurewebsites.net/probability

```
{
        "CANTIDAD_POLIZAS": "0",
        "YEARS_VIGENTE1": "0",
        "RAMOS": "2",
        "NOTA1": "OTROS",
        "TOTAL_SINIES_DESDE2005": "5",
        "COD_SEGMENTO_1": "170",
        "SOCRENTAMEDIA": "16553.0",
        "DENSIDAD":  "1.197",
        "HABITANTES": "65000.0",
        "CLASIFICACION_EMPLEADO": "NO EMPLEADO"
}
```

Y la respuesta obtenida:

```
{
   "code": 4,
   "type": "ok",
   "message": "0"
}
```

El código Python:

```
def probablity(cantidad_polizas ,
            years_vigente ,
            ramos ,
            nota1 ,
            total_sinies_desde2005 ,
            cod_segmento ,
            socrentamedia ,
            densidad ,
            habitantes ,
            clasificacion_empleado ,
            socsegmentoseguros):
        url = 'https://productobigdataml.azurewebsites.net/probability'
        parameters = {'CANTIDAD_POLIZAS':cantidad_polizas,
                      'YEARS_VIGENTE1':years_vigente1,
                      'RAMOS':ramos,
                      'NOTA1':nota1,
                      'TOTAL_SINIES_DESDE2005':total_sinies_desde2005,
                      'COD_SEGMENTO_1':cod_segmento,
                      'SOCRENTAMEDIA':socrentamedia,
                      'DENSIDAD':densidad,
                      'HABITANTES':habitantes,
                      'CLASIFICACION_EMPLEADO':clasificacion_empleado}
        resp = requests.post(url, json=parameters)
        message = resp.json()['message']
        code = resp.json()['code']
        type = resp.json()['type']
        return str(message)
```

Como se puede apreciar, esto nos devuelve tres variables. Message, code y type. Realmente la que nos interesa rescatar es message que es la que contiene la clave para continuar con la siguiente validación. De ahí que el return solo se haga sobre este dato.

El dato que nos devuelve es un 0 o un 1 que decidirá si vamos a ejecutar el algoritmo de positivos o de negativos.

Algoritmo POSITIVOS

Cuando el algoritmo Probability nos devuelve un 1 en la variable “message”, ejecutaremos la llamada al algoritmo de positivos. Esto significa que el importe que nos devolverá será siempre un importe positivo.

Variables de entrada:

https://productobigdataml.azurewebsites.net/positive

```
{
"COD_SEGMENTO_1": "100",
"CANTIDAD_POLIZAS": "2",
"YEARS_VIGENTE1": "3",
"TOTAL_SINIES_DESDE2005": "2",
"COD_PROVINCIA": "28",
"NOTA1": "OTROS",
"NOTA2V": "1",
"RAMOS": "1",
"CONT_VIGOR": "1",
"SOCGRUPOMOSAIC": "J ",
"SOCSEGMENTOSEGUROS": "TA2",
"EDAD": "34"
}
```

Y la respuesta:

```
{
   "code": 4,
   "type": "ok",
   "message": "2831"
}
```

El código Python:

```
def positivos(cod_segmento_1 ,
              cantidad_polizas ,
              years_vigente1 ,
              total_sinies_desde2005 ,
              cod_provincia ,
              nota1 ,
              nota2v ,
              ramos ,
              cont_vigor ,
              socgrupomosaic ,
              socsegmentoseguros ,
              edad):
        url = 'https://productobigdataml.azurewebsites.net/positive'
        parameters = {'COD_SEGMENTO_1':cod_segmento,
                      'CANTIDAD_POLIZAS':cantidad_polizas,
                      'YEARS_VIGENTE1':years_vigente1,
                      'TOTAL_SINIES_DESDE2005':total_sinies_desde2005,
                      'COD_PROVINCIA':cod_provincia,
                      'NOTA1':nota1,
                      'NOTA2V':nota2v,
                      'RAMOS':ramos,
                      'CONT_VIGOR':cont_vigor,
                      'SOCGRUPOMOSAIC':socgrupomosaic,
                      'SOCSEGMENTOSEGUROS':socsegmentoseguros,
                      'EDAD':edad}
        resp = requests.post(url, json=parameters)
        message = resp.json()['message']
        code = resp.json()['code']
        type = resp.json()['type']
        return str(message)
```

El algoritmo también nos devuelve tres variables. Message, code y type. Realmente la que nos interesa rescatar es message que es la que nos devuelve el importe que es el margen simulado.

Como anteriormente, nos quedamos con la variable message que es la que nos da el valor para después calcular el importe optimizado a partir del factor y calculando su Scoring.

Algoritmo NEGATIVOS

Cuando el algoritmo Probability nos devuelve un 0 en la variable "message", ejecutaremos la llamada al algoritmo de negativos. Esto significa que el importe que nos devolverá será siempre un importe negativo.

Variables de entrada:

```
https://productobigdataml.azurewebsites.net/negative
{
"TOTAL_SINIES_DESDE2005": "0",
"CANTIDAD_POLIZAS": "1",
"COD_SEGMENTO_1": "100",
"COD_PROVINCIA": "28",
"YEARS_VIGENTE1": "1",
"SOCGRUPOMOSAIC": "J ",
"RAMOS": "1",
"DIAS_TMAX_30": "100",
"SOCPAROTOT": "100",
"NOTA2V": "1",
"EDAD": "36",
"NOTA1": "OTROS",
"SOCSEGMENTOSEGUROS": "TA2",
"R_MAX_VEL": "100",
"TMIN": "100",
"DTORMENTA": "100",
"DP100": "100",
"DNIEBLA": "100"
}
```

Y la respuesta:

```
{
   "code": 4,
   "type": "ok",
   "message": "-3992"
}
```

El código Python:

```
def negativo(total_sinies_desde2005 ,
            cantidad_polizas ,
            cod_segmento ,
            cod_provincia ,
            years_vigente ,
            socgrupomosaic ,
            ramos ,
            dias_tmax_30 ,
            socparotot ,
            nota2v ,
            edad ,
            nota1 ,
            socsegmentoseguros ,
            r_max_vel ,
            tmin ,
            dtormenta ,
            dp100 ,
            dniebla):
        url = 'https://productobigdataml.azurewebsites.net/negative'
        parameters = {'TOTAL_SINIES_DESDE2005':total_sinies_desde2005,
                      'CANTIDAD_POLIZAS':cantidad_polizas,
                      'COD_SEGMENTO_1':cod_segmento,
                      'COD_PROVINCIA':cod_provincia,
                      'YEARS_VIGENTE1':years_vigente1,
                      'SOCGRUPOMOSAIC':socgrupomosaic,
                      'RAMOS':ramos,
                      'DIAS_TMAX_30':dias_tmax_30,
                      'SOCPAROTOT':socparotot,
                      'NOTA2V':nota2v,
                      'EDAD':edad,
                      'NOTA1':nota1,
                      'SOCSEGMENTOSEGUROS':socssegmentoseguros,
                      'R_MAX_VEL':r_max_vel,
                      'TMIN':tmin,
                      'DTORMENTA':dtormenta,
                      'DP100':dp100,
                      'DNIEBLA':dniebla}
        resp = requests.post(url, json=parameters)
        message = resp.json()['message']
        code = resp.json()['code']
        type = resp.json()['type']
        return str(message)
```

Al igual que en los otros algorítmos, nos devuelve las mismas 3 variables, message, code y type. De nuevo la que nos interesa rescatar es message que es la que nos devuelve el valor para después calcular el importe optimizado a partir del factor y calculando su Scoring.

Algoritmo NOCLIENTE

Cuando el cliente no pertenece a Reale, se ejecuta este algoritmo.

Este ofrece una simulación nueva, ya que no contiene ningún dato a rescatar para aplicar ningún factor. Esto es así, porque no existe información en la BBDD de Reale. A partir de este momento, el cliente pasara a estar dado de alta en nuestro sistema.

Variables de entrada

https://productobigdataml.azurewebsites.net/noclient

```
{
"NOTA1": "OTROS",
"NOTA2V": "1",
"COD_PROVINCIA": "28",
"EDAD": "40",
"SOCGRUPOMOSAIC": "J ",
"SOCPAROTOT": "100",
"SOCURBANIDAD": "100",
"SOCSEGMENTOSEGUROS": "TA2",
"SOCRENTAMEDIA": "100"
 }
```

La salida:


```
{
  "code": 4,
  "type": "ok",
  "message": "237"
}
```


El código Python:

```
def noclientes(nota1 ,
               nota2v ,
               cod_provincia ,
               edad ,
               socgrupomosaic ,
               socparotot ,
               socurbanidad ,
               socsegmentoseguros ,
               socrentamedia):
        url = 'https://productobigdataml.azurewebsites.net/noclient'
        parameters = {'NOTA1':nota1,
                      'NOTA2V':nota2v,
                      'COD_PROVINCIA':cod_provincia,
                      'EDAD':edad,
                      'SOCGRUPOMOSAIC':socgrupomosaic,
                      'SOCPAROTOT':socparotot,
                      'SOCURBANIDAD':socurbanidad,
                      'SOCSEGMENTOSEGUROS':socsegmentoseguros,
                      'SOCRENTAMEDIA':socrentamedia}
        resp = requests.post(url, json=parameters)
        message = resp.json()['message']
        code = resp.json()['code']
        type = resp.json()['type']
        return str(message)
```

De nuevo, obtenemos tres variables. Message, code y type. Realmente la que nos interesa rescatar es message que es la que nos devuelve el importe que es el margen simulado

Como anteriormente, nos quedamos con la variable message que es la que nos da el valor para después calcular el importe optimizado a partir del factor y calculando su Scoring.

6.6 Despliegue

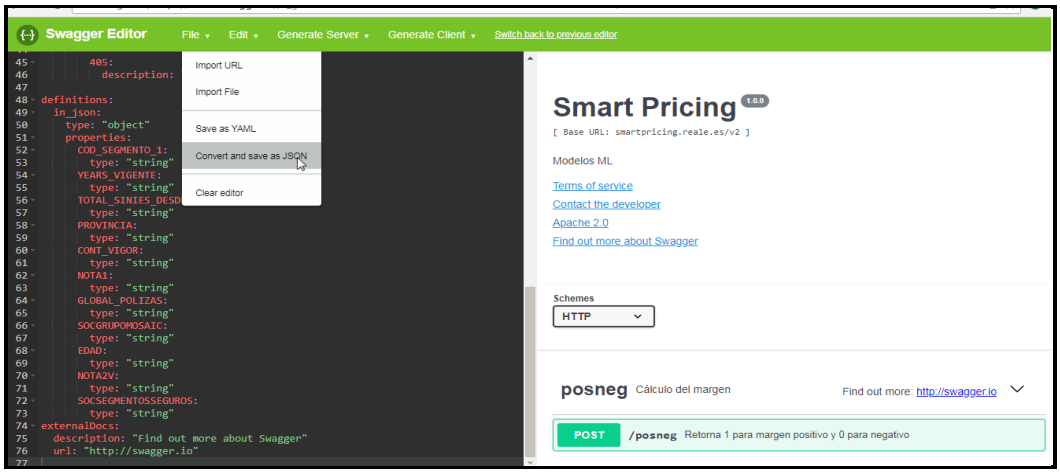
Tras la evaluación y la conclusión de la estrategia a implementar, se procede al despliegue. Esto es un resumen de los pasos necesarios e instrucciones para llevarlo a cabo.

6.6.1 Generación/despliegue de la API para publicar Modelos Machine Learning

Se utiliza Swagger para la creación del API. Los pasos son los siguientes:

1. Se crea en el editor online, en lenguaje Yaml, la definición de todas las variables, que va a contener la API.

2. Se convierte a JSON



3. Se genera el server

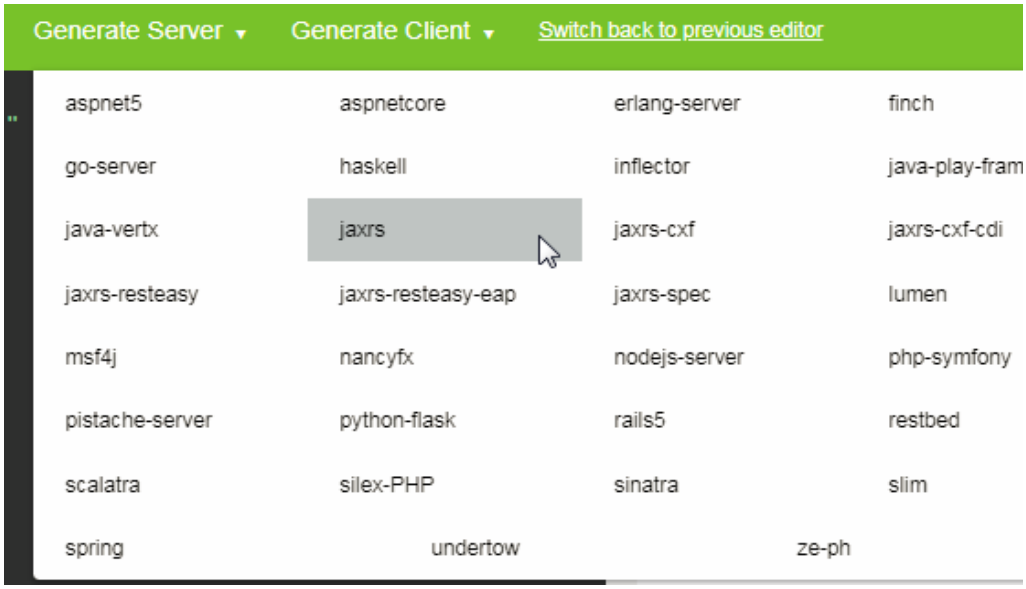


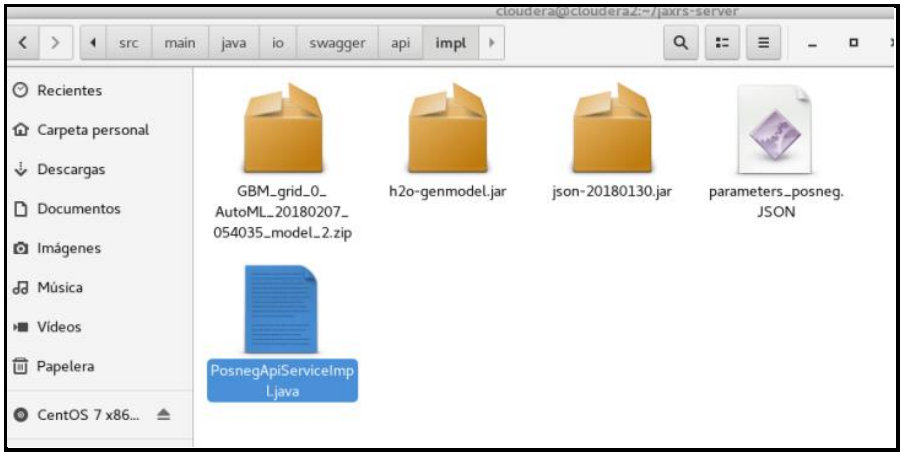
Figura 49. Generación Server

Esto genera un fichero .zip con el esqueleto del API.

4. Se descomprime dicho fichero en la ruta de trabajo

5. Se debe encontrar la ruta de la implementación del método para actualizar la biblioteca de Maven con las clases de h2o. En nuestro caso dicha ruta es la siguiente:

C:\VM_NAS\Developer\Swagger\jaxrs-server\src\main\java\io\swagger\api\impl



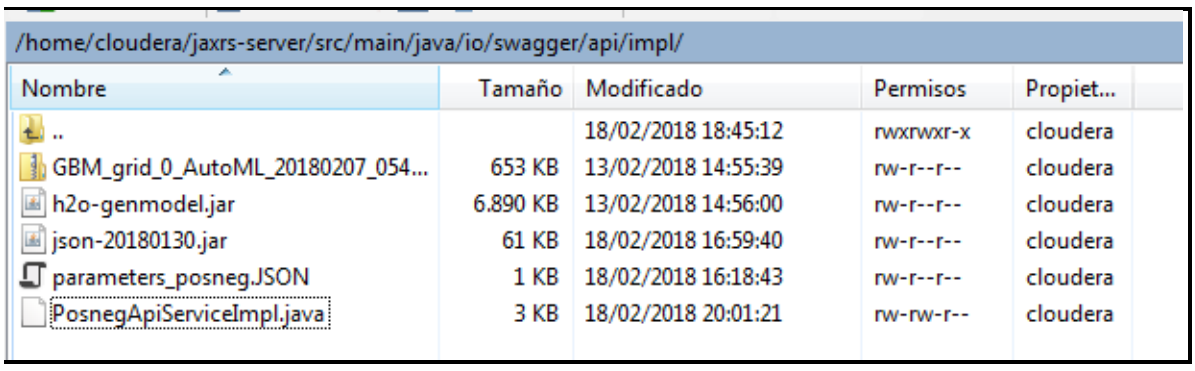
En esta ruta se copia la librería de **h2o-genmodel.jar.** Para incorporarla posteriormente en la biblioteca de Maven, lo que es requerido para compilar el Proyecto, se ejecuta el siguiente comando:

```
mvn install:install-file -Dfile=C:\VM_NAS\Developer\Swagger\jaxrs-server\src\main\java\io\swagger\api\impl/h2o-genmodel.jar -DgroupId=hex.genmodel -DartifactId=h2o-genmodel -Dversion=1.0.0 -Dpackaging=jar -DgeneratePom=true
```

Esto se hace solo una vez, de forma que queda actualizada las clases de h2o en la biblioteca de Maven (una vez realizado este paso, el fichero h2o-genmodel.jar se podría eliminar o mover a otro sitio y no pasaría nada).

6. Método de despliegue de un modelo exportado como POJO

- Swagger ha generado un fichero para el método a implementar, llamado PosnegApiServiceImpl.ja en la ruta "C:\VM_NAS\Developer\Swagger\jaxrs-server\src\main\java\io\swagger\api\impl"



- Editamos fichero y reemplazamos por el siguiente código:

```
package io.swagger.api.impl;
import io.swagger.api.*;
import io.swagger.model.*;
import io.swagger.model.InJson;
import java.util.List;
import io.swagger.api.NotFoundException;
import java.io.InputStream;
import org.glassfish.jersey.media.multipart.FormDataContentDisposition;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.SecurityContext;
import javax.validation.constraints.*;
import java.io.*;
import hex.genmodel.easy.RowData;
import hex.genmodel.easy.EasyPredictModelWrapper;
import hex.genmodel.easy.prediction.*;
import hex.genmodel.MojoModel;

 import hex.genmodel.MojoReaderBackendFactory;
 import static hex.genmodel.MojoReaderBackendFactory.CachingStrategy;
 import hex.genmodel.MojoReaderBackend;
 import hex.genmodel.ModelMojoReader;

 import java.net.URL;

@javax.annotation.Generated(value = "io.swagger.codegen.languages.JavaJerseyServerCodegen", date = "2018-02-18T16:59:59.845Z")
public class PosnegApiServiceImpl extends PosnegApiService {
        public String resultado;

        private static String modelClassName = "GBM_grid_0_AutoML_20180207_054035_model_2";
        @Override
        public Response posneg(InJson inJson, SecurityContext securityContext) throws NotFoundException {
                try {
                rawModel = (hex.genmodel.GenModel) Class.forName(modelClassName).newInstance();
                EasyPredictModelWrapper model = new EasyPredictModelWrapper(rawModel);

                        RowData row = new RowData();
                        row.put("COD_SEGMENTO_1", inJson.getCoDSEGMENTO1());
                        row.put("YEARS_VIGENTE", inJson.getYEARSVIGENTE());
                        row.put("TOTAL_SINIES_DESDE2005", inJson.getToTALSINIESDESDE2005());
                        row.put("PROVINCIA",inJson.getPROVINCIA());
                        row.put("CONT_VIGOR", inJson.getCONTVIGOR());
                        row.put("NOTA1", inJson.getNoTA1());
                        row.put("GLOBAL_POLIZAS", inJson.getGLOBALPOLIZAS());
                        row.put("SOCGRUPOMOSAIC", inJson.getSOCGRUPOMOSAIC());
                        row.put("EDAD", inJson.getEDAD());
                        row.put("NOTA2V", inJson.getNoTA2V());
                        row.put("SOCSEGMENTOSEGUROS", inJson.getSOCSEGMENTOSSEGUROS());

                        BinomialModelPrediction p = model.predictBinomial(row);
                        resultado=p.label;
                }catch(ClassNotFoundException e) {
                }catch(InstantiationException e) {
                }catch(hex.genmodel.easy.exception.PredictException e) {
                        System.err.println("Caught IOException: " + e.getMessage());
                }
                return Response.ok().entity(new ApiResponseMessage(ApiResponseMessage.OK, resultado)).build();
        }
}
```

- Copiar fichero del modelo exportado en POJO:

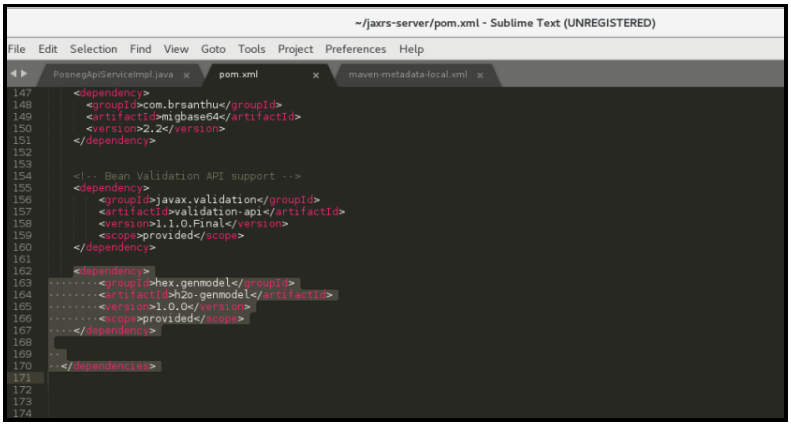
"GBM_grid_0_AutoML_20180207_054035_model_2.java"

en la ruta

"C:\VM_NAS\Developer\Swagger\jaxrs-server\src\main\java "

Al copiarse en esa ruta, cuando se genere el fichero war, ya maven lo incorporará, porque esa ruta está indicada en el fichero pom.xml

- Modificar el fichero pom.xml que está en la ruta inicial del proyecto



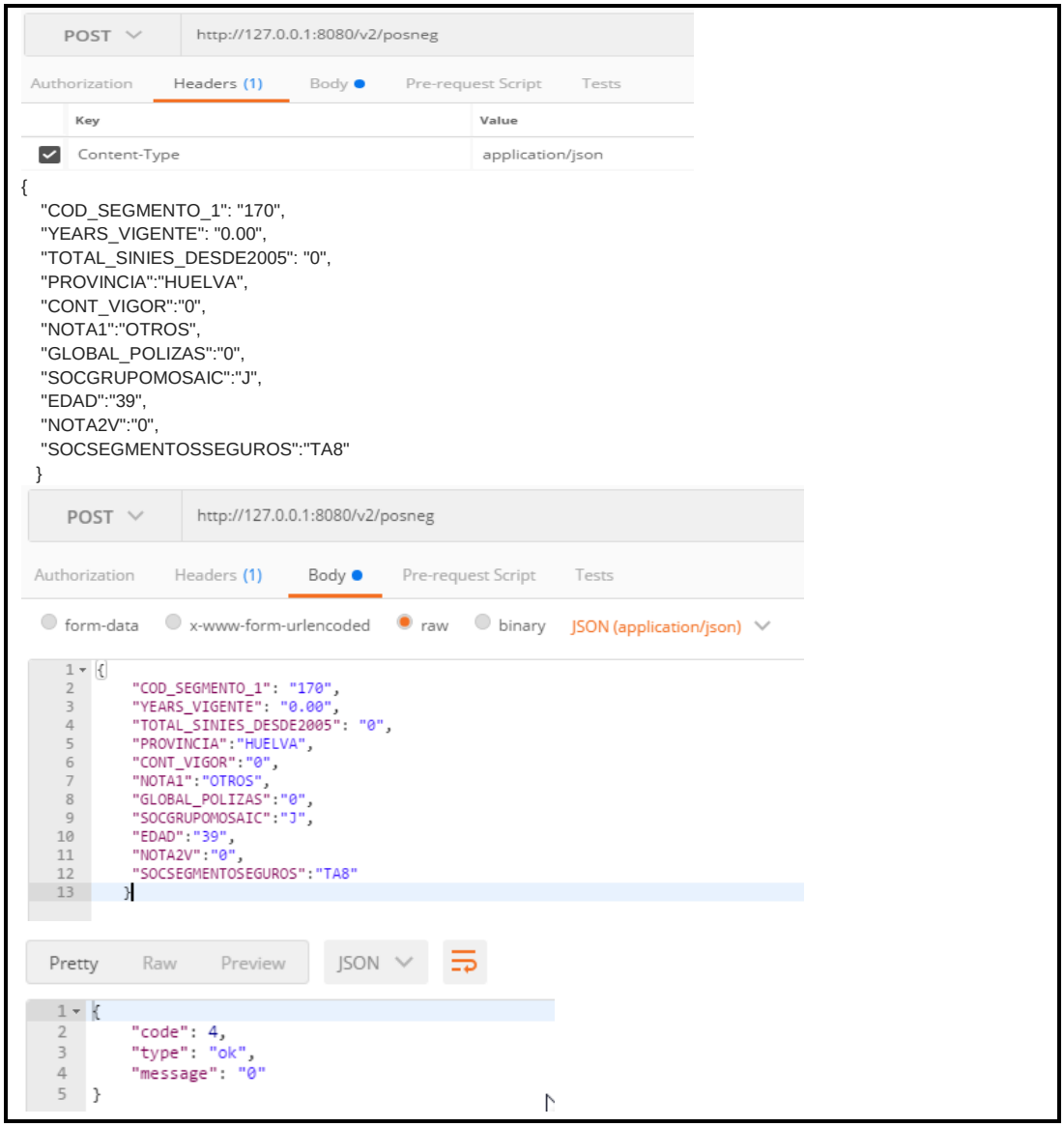
Se adiciona en dependencia:

```
<!-- Modelo Machine Learning -->
 <dependency>
      <groupId>hex.genmodel</groupId>
      <artifactId>h2o-genmodel</artifactId>
      <version>1.0.0</version>
   </dependency>


Se elimina  de dependencia anterior javax.validation linea de scope
<dependency>
      <groupId>javax.validation</groupId>
      <artifactId>validation-api</artifactId>
      <version>1.1.0.Final</version>
   </dependency>


 </dependencies>
```

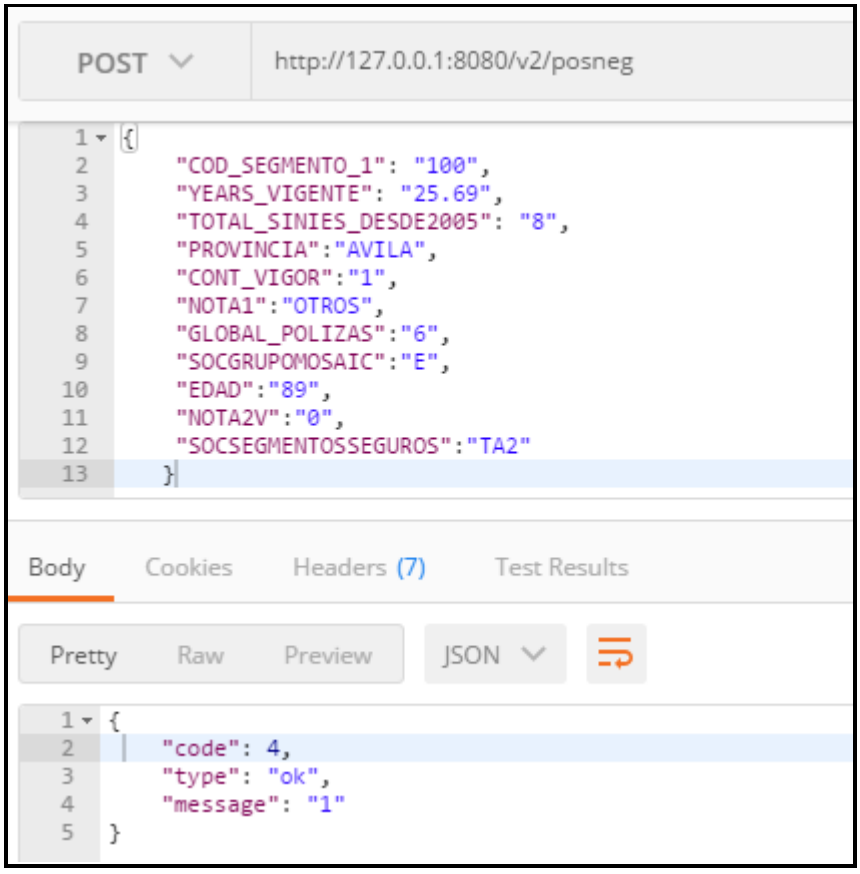
- Para compilar el paquete vamos a la ruta:C:\VM_NAS\Developer\Swagger\jaxrs-server"

  Y ejecutamos el comando: “mvn package jetty:run”

  Para ejecutar la automatización de pruebas sobre la API, se utiliza el programa Postman. Luego se abre el postman y se realizan los test:

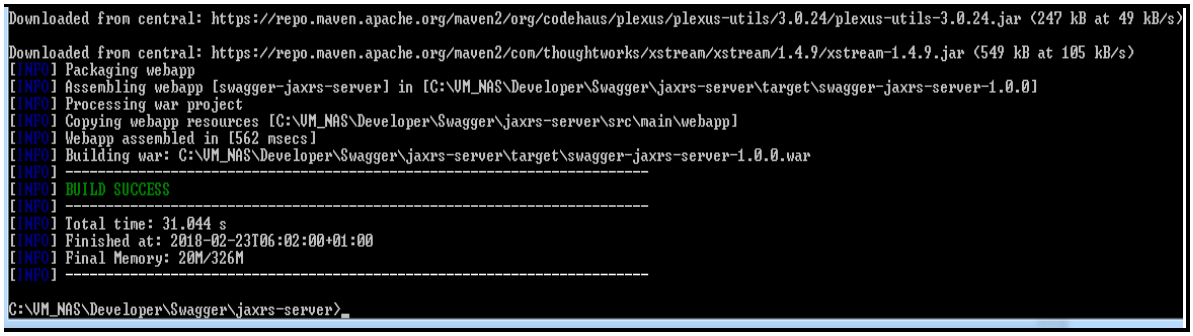


Test que retorna 1



- Realizados los test, se continua con el despliegue para lo que se genera el .war ejecutando desde la misma ruta "C:\VM_NAS\Developer\Swagger\jaxrs-server" el comando:

"mvn package war:war"

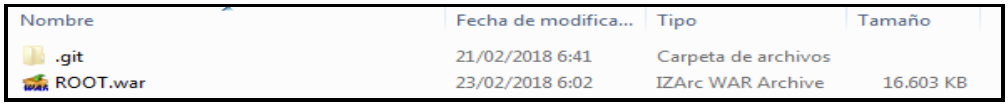


Se renombra entonces el fichero .war generado en la carpeta Target como ROOT.war:

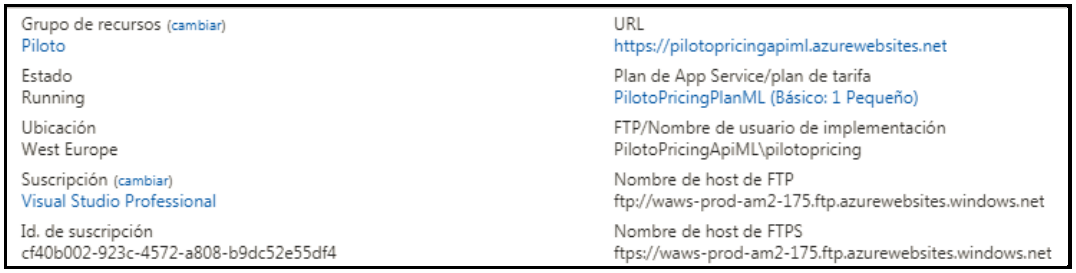
"rename swagger-jaxrs-server-1.0.0.war ROOT.war"

Para desplegar en Azure, se copia el fichero ROOT.war en la carpeta que se utilizará para dicho despliegue:

"C:\VM_NAS\Developer\Git\PilotoPricing"



En AZURE se crea una "App Service" para publicar el "war", esta es la manera de asignarle procesamiento al modelo.

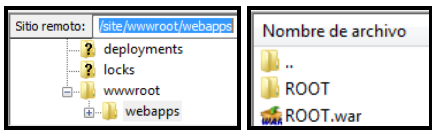


Para subir el fichero se utiliza el programa de FTP FileZilla, los datos necesarios son:

Servicio ftp: waws-prod-am2-183.ftp.azurewebsites.windows.net
User: productobigdataml\pilotopricing / XXXXXXXX

El fichero finalmente se copia en la ruta:

"/site/wwwroot/webapps"



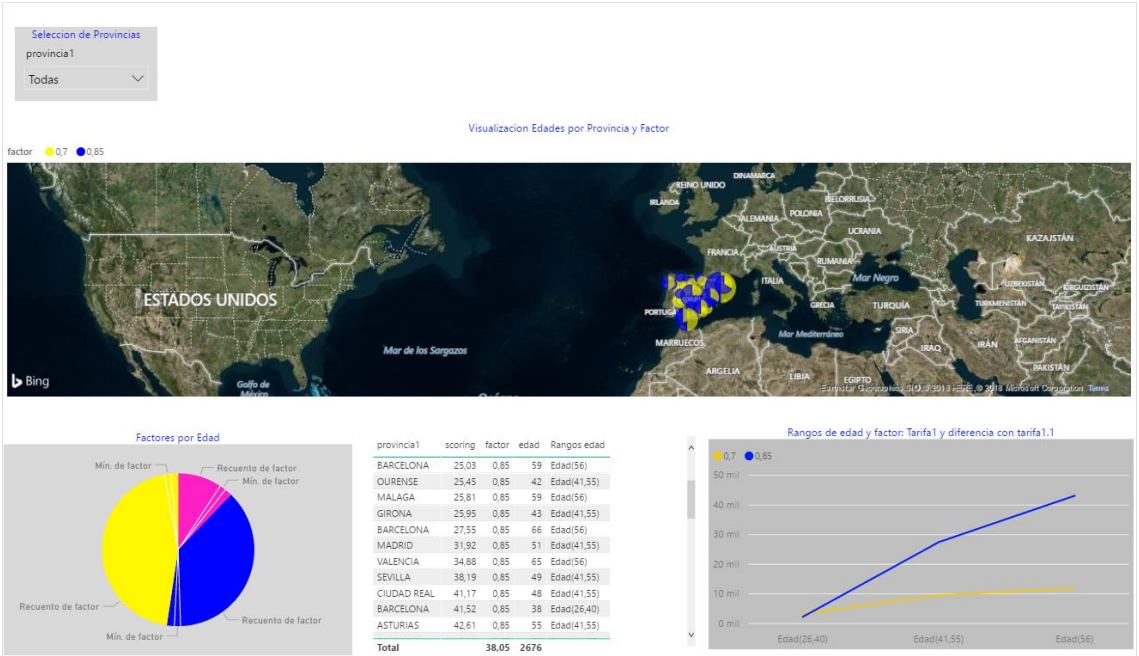
7. Resultados

7.1 Visualización de los Datos

Como se explica en capítulos anteriores. Se genera una Rest Api para generar un fichero con formato óptimo para hacer la importación a Power Bi.

El set de datos está formado por aquellas pruebas que hemos ido realizando para comprobar que la aplicación funcionase de forma correcta. Es decir, no existen demasiados datos y al ser un piloto, tanto esta parte de la documentación como los KPI, deberían volverse a generar una vez pasara un tiempo razonable de prueba. Es decir, al menos unos meses después de haber puesto el piloto en producción.

Tenemos varios tipos de visualizaciones. Todas se basan en el factor que se aplica a cada uno de los clientes, que es el % que se aplica para el cálculo del Scoring que finalmente se aplicara a la nueva simulación.



provincia1	scoring	factor	edad	Rangos edad
BARCELONA	25,03	0,85	59	Edad(56)
OURENSE	25,45	0,85	42	Edad(41,55)
MALAGA	25,81	0,85	59	Edad(56)
GIRONA	25,95	0,85	43	Edad(41,55)
BARCELONA	27,55	0,85	66	Edad(56)
MADRID	31,92	0,85	51	Edad(41,55)
VALENCIA	34,88	0,85	65	Edad(56)
SEVILLA	38,19	0,85	49	Edad(41,55)
CIUDAD REAL	41,17	0,85	48	Edad(41,55)
BARCELONA	41,52	0,85	38	Edad(26,40)
ASTURIAS	42,61	0,85	55	Edad(41,55)
Total		38,05	2676	

Figura 50. Visualización Datos 1

El primer grafico es para realizar una segmentación de datos. Podríamos seleccionar una provincia y que el mapa muestre solo los datos de ese lugar, igual con el resto de las imágenes.

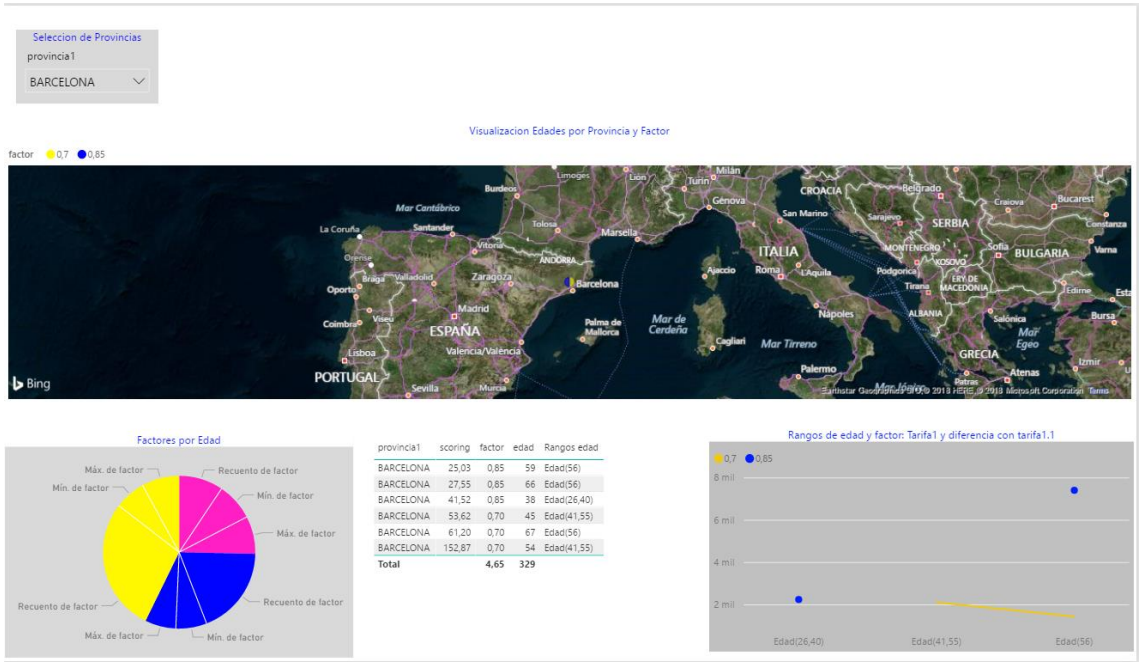


Figura 51. Visualización Datos 2

Con respecto a la información que vemos en la visualización del mapa, si nos colocamos encima de la burbuja de la ciudad seleccionada, podemos ver información añadida. En este caso, a parte de la provincia a la que pertenece, también el factor que hemos seleccionado y cuantas personas existen dentro del rango de edades que hemos definido.

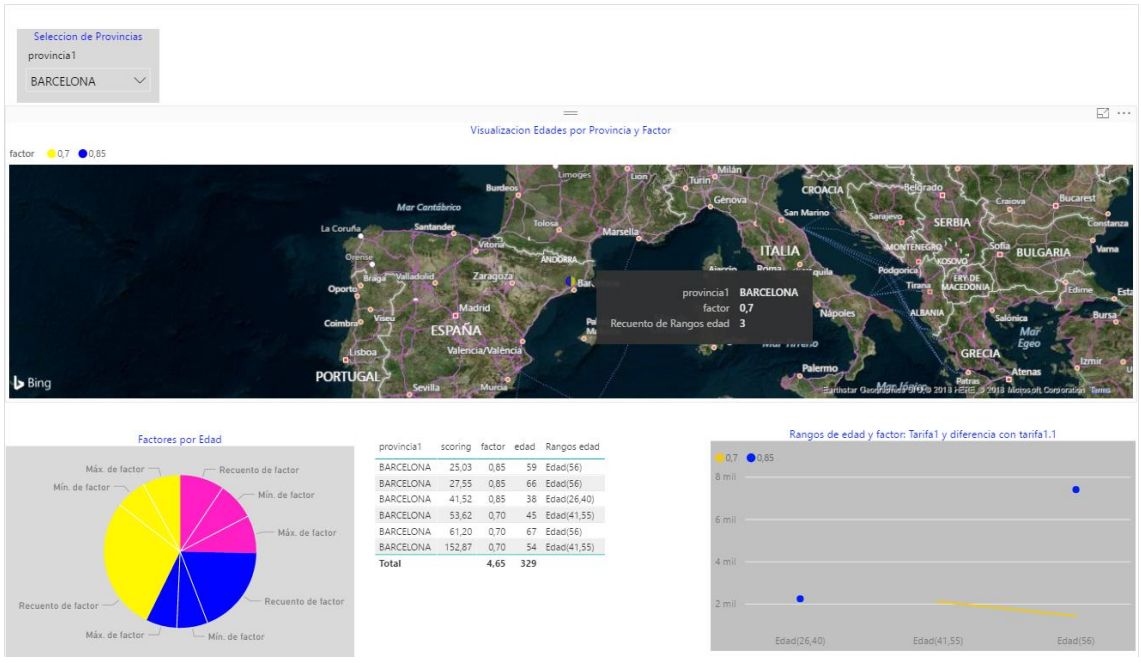
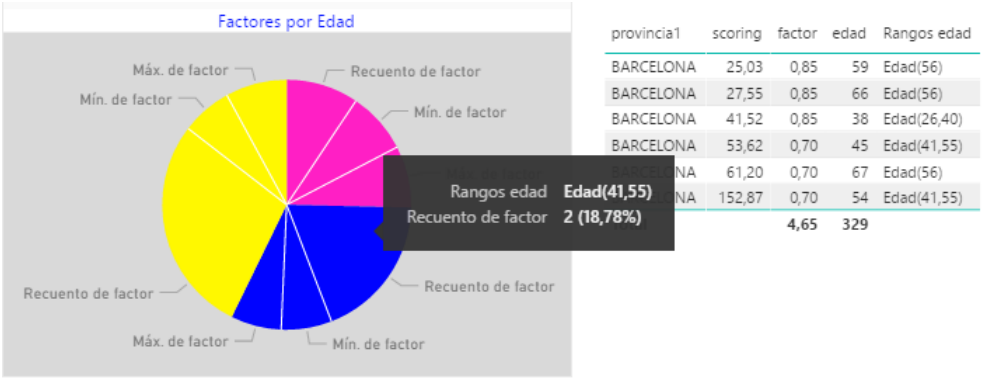


Figura 52. Visualización Datos 3

Con respecto al grafico Circular, estamos informando sobre Factores por Edad. Es decir…a simple vista vemos el % relacionado con la cantidad de personas que existen de diferentes rangos de edad.

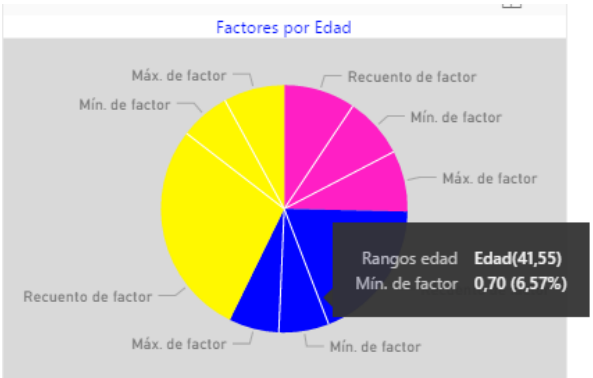
También en este gráfico, si nos colocamos encima, podremos ver por un lado el rango de edad que estamos consultando y cuantas personas tienen aplicado ese factor.



provincia1	scoring	factor	edad	Rangos edad
BARCELONA	25,03	0,85	59	Edad(56)
BARCELONA	27,55	0,85	66	Edad(56)
BARCELONA	41,52	0,85	38	Edad(26,40)
BARCELONA	53,62	0,70	45	Edad(41,55)
BARCELONA	61,20	0,70	67	Edad(56)
NA	152,87	0,70	54	Edad(41,55)
		4,65	329	

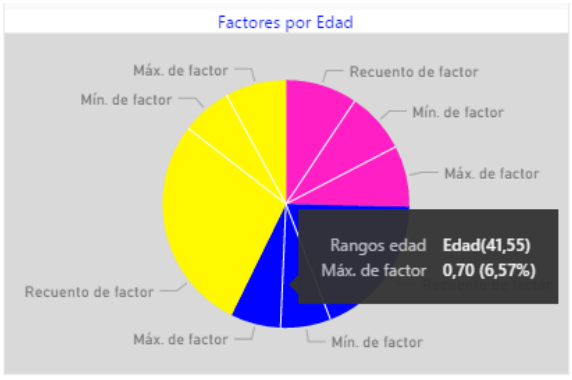
Figura 53 Detalle 1 Visualización Datos

Si ponemos el ratón en las otras particiones dentro del mismo rango de edad, nos dará a parte del rango de edad, el factor que se está aplicando, en caso de haber factores diferentes. En este caso los dos factores son iguales, por eso muestra dos porciones del grafico iguales.



provincia1	scoring	factor	edad	Rangos edad
BARCELONA	25,03	0,85	59	Edad(56)
BARCELONA	27,55	0,85	66	Edad(56)
BARCELONA	41,52	0,85	38	Edad(26,40)
BARCELONA	53,62	0,70	45	Edad(41,55)
BARCELONA	61,20	0,70	67	Edad(56)
BARCELONA	152,87	0,70	54	Edad(41,55)
Total		4,65	329	

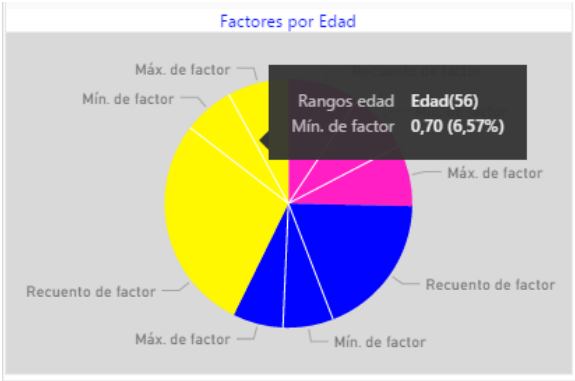
Figura 54. Detalle 2 Visualización Datos



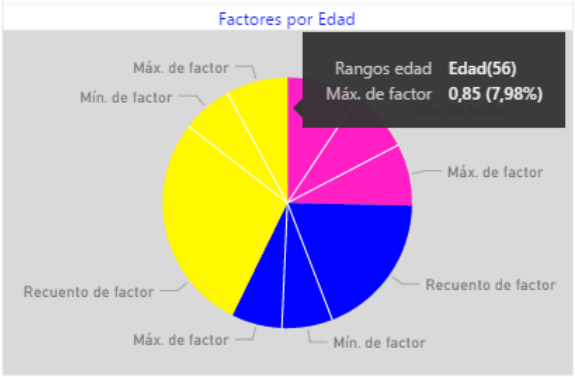
provincia1	scoring	factor	edad	Rangos edad
BARCELONA	25,03	0,85	59	Edad(56)
BARCELONA	27,55	0,85	66	Edad(56)
BARCELONA	41,52	0,85	38	Edad(26,40)
BARCELONA	53,62	0,70	45	Edad(41,55)
BARCELONA	61,20	0,70	67	Edad(56)
BARCELONA	152,87	0,70	54	Edad(41,55)
Total		**4,65**	**329**	

Figura 55. Detalle 3 Visualización Datos

En la porción amarilla que pertenece al rango de edades 56 existen dos rangos, y como se puede apreciar, uno es de mayor dimensión que el otro. Veremos que cada uno pertenece a un rango diferente.



provincia1	scoring	factor	edad	Rangos edad
BARCELONA	25,03	0,85	59	Edad(56)
BARCELONA	27,55	0,85	66	Edad(56)
BARCELONA	41,52	0,85	38	Edad(26,40)
BARCELONA	53,62	0,70	45	Edad(41,55)
BARCELONA	61,20	0,70	67	Edad(56)
BARCELONA	152,87	0,70	54	Edad(41,55)
Total		**4,65**	**329**	

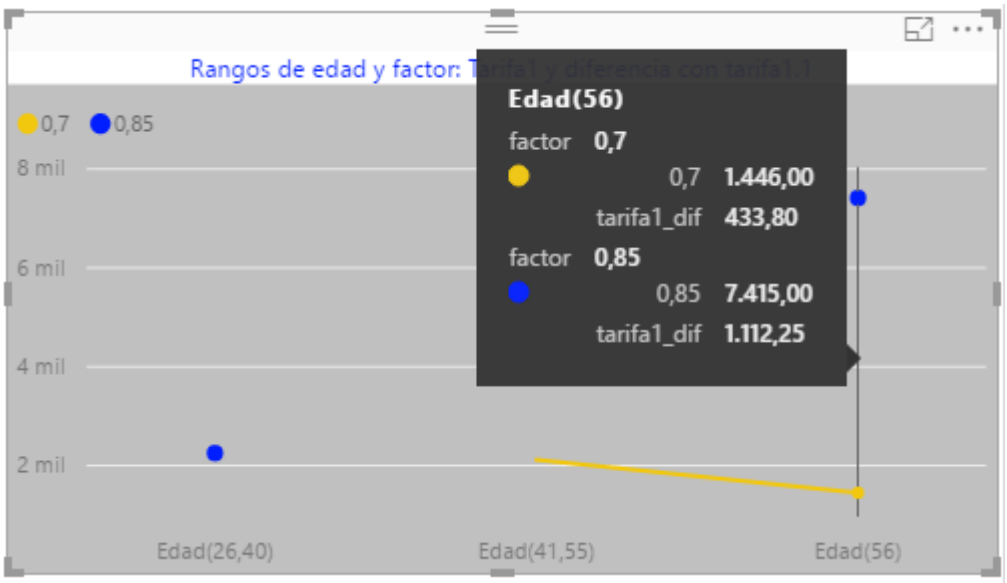


provincia1	scoring	factor	edad	Rangos edad
BARCELONA	25,03	0,85	59	Edad(56)
BARCELONA	27,55	0,85	66	Edad(56)
BARCELONA	41,52	0,85	38	Edad(26,40)
BARCELONA	53,62	0,70	45	Edad(41,55)
BARCELONA	61,20	0,70	67	Edad(56)
BARCELONA	152,87	0,70	54	Edad(41,55)
Total		**4,65**	**329**	

Figura 56. Detalle 4 Visualización Datos

La tabla de contenido que se muestra a continuación del grafico circular, no necesita mucha explicación, es simplemente a modo informativo mostrar algunos datos que nos dan visibilidad para poder comprobar que lo que estamos visualizando es correcto.

Para el ultimo el grafico de líneas se visualiza, dentro del rango de edad que escogemos, tanto los factores diferentes que existen, como la tarifa aplicada por el tarificador original de Reale y la diferencia con respecto a la tarifa que se calcula con los nuevos algoritmos.



Gráfica 9. Detalle 5 Visualización Datos

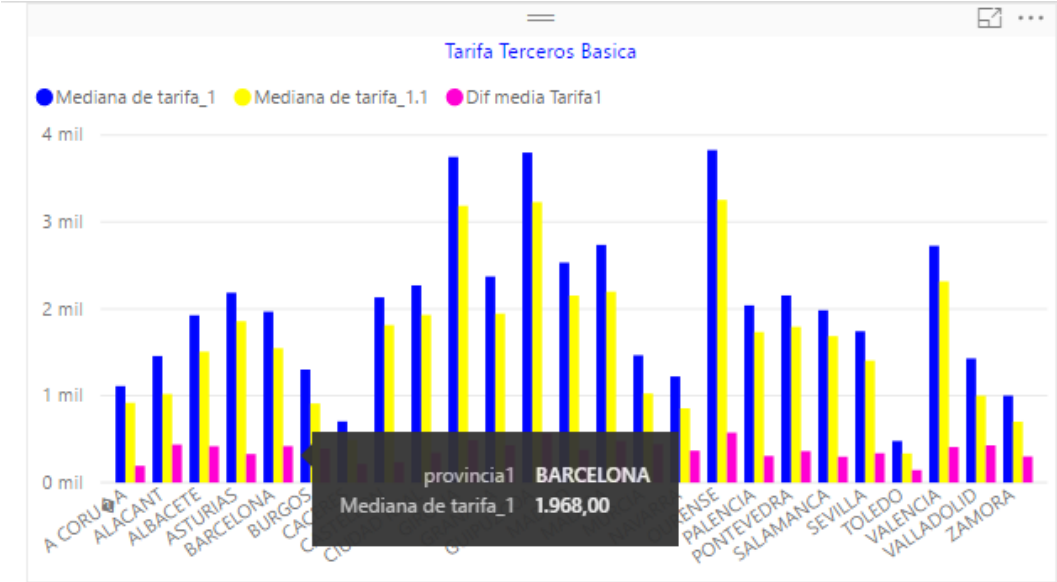
Abordamos ahora la parte de visualización centrada en las tarifas.

Gráfica 10. Gráficas Visualización Tarifas Generadas

Se hace un estudio sobre las tarifas generadas actualmente por el tarificador de Reale y aplicando los algoritmos de los que se habla en capítulos anteriores, se genera la nueva tarifa optimizada.

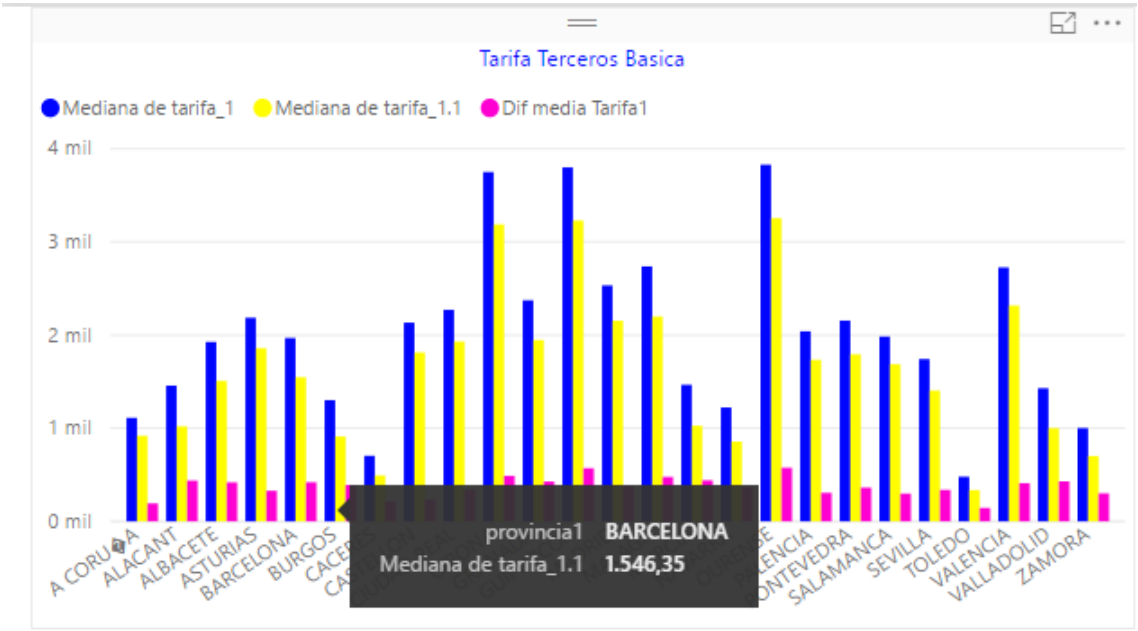
Como se puede comprobar en la visualización, todos los precios optimizados tienen un valor con un 12% mínimo de mejora en la simulación obtenida con respecto al valor inicial que ofrece el tarificador de Reale. Debemos observar que estos datos pertenecen a las medias obtenidas por provincias.

Obtenemos más información si nos posicionamos sobre las barras. Podemos saber el precio medio sobre la tarifa original en azul.



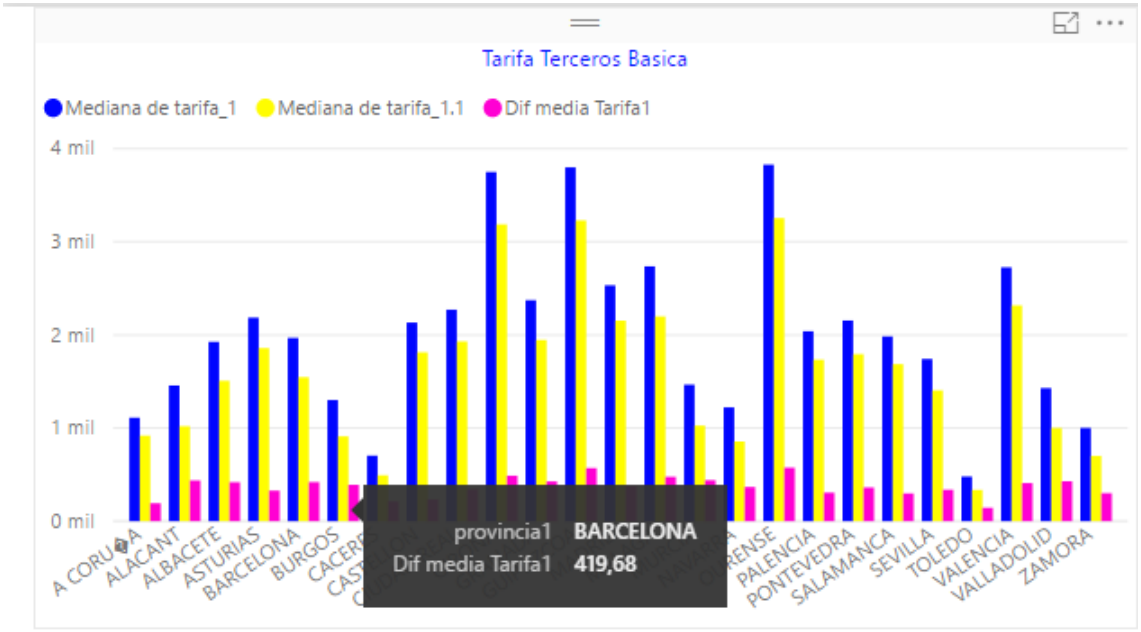
Gráfica 11. Precio medio Tarifa Terceros Básica Original por Provincia.

El precio medio sobre la tarifa optimizada en amarillo.



Gráfica 12. Precio medio Tarifa Terceros Básica Optimizada

Y finalmente en color morado, la media sobre la diferencia de las dos tarifas anteriores.



Gráfica 13. Media de la diferencia de las Tarifas Terceros Básica Original y Optimizada

8. Costes

En este apartado presentamos los costes referidos a los que se calculan para los Recursos Humanos empleados. Este proyecto no generó ningún coste adicional puesto que se utilizaron los recursos ya disponibles de la empresa Reale.

	Rol 1/horas	Rol 2/horas	Rol 3/horas
Investigación y documentación	213	213	213
Entendimiento del Negocio y Gestión del Desarrollo	17	17	17
Extracción inicial de datos	34	34	34
Entendimiento de los datos	51	51	51
Preparación de los datos	26	26	26
Modelización y Desarrollo	34	34	34
Evaluación	21	21	21
Despliegue Azure	21	21	21
Presentación Resultados	9	9	9
Total horas/Rol	427	427	427
Total €/Rol	16.000 €	16.000 €	16.000 €
Total € Proyecto		48.000 €	

Tabla 24. Costes de Proyecto.

9. Conclusiones

El piloto generado, se centra en dar una visión sobre los objetivos de mejora de la experiencia de usuario y la generación de una plataforma que sea más sencilla de utilizar, así como la consecución del mejor margen generado para Reale como empresa y la mejor simulación de precios para el cliente que utiliza nuestra plataforma.

Somos conscientes de que es necesario poner el piloto en producción para conseguir datos fiables sobre la valoración que el usuario da a nuestra nueva propuesta.

Con respecto al equipo de trabajo. A pesar de que teníamos larga experiencia en generación de proyectos ya que todas veníamos del mundo de la informática, la diferencia de perfiles hace que las fases más dedicadas a la definición de los requisitos sean por primera vez aplicadas y utilizadas para parte del equipo.

La consecución de este proyecto nos ha proporcionado una visión total, desde la toma de requisitos, generación de KPI's, realización cronogramas, hasta la parte de programación y testeo. Esto ha generado la necesidad de realización de muchas reuniones y puestas en común sobre cómo se deberían realizar este tipo de tareas.

La parte de infraestructura ha sido un punto que ha generado bastante inseguridad, dado que anteriormente no nos habíamos visto obligados a participar en esta fase de los proyectos y nuestro único conocimiento se refiere a los obtenidos en la escuela.

Finalmente, y desde la visión del trabajo finalizado, tenemos la certeza de que se han conseguido aplicar muchas de las herramientas y conocimientos proporcionados por los estudios realizados, así como otros que a pesar de no estar dentro de los conocimientos impartidos, se han aplicado a dicho piloto.

Esta ha sido la primera toma de contacto con este tipo de proyectos, y nos aporta una experiencia que consideramos necesaria para incorporarnos a este mercado laboral.

10. Mejoras y Líneas Futuras

Como todo piloto, el apartado de mejoras es algo a tener en cuenta. En la lluvia de ideas se suelen abordar objetivos que finalmente y básicamente por los tiempos impuestos a los proyectos, deben ser retrasados y abordados en fases posteriores.

Para la FASE II de este piloto tenemos varios objetivos:

1.- Añadir la simulación para seguros de Hogar que actualmente no se está teniendo en cuenta.

2.- Añadir el análisis de imagen. El objetivo es añadir la capacidad a la herramienta de mediante la utilización de una foto, extraer los datos necesarios que actualmente se teclean en la aplicación. Dando así las dos opciones al usuario.

También esta técnica, nos permite realizar un tratamiento nuevo para la realización de trámites de siniestros, tanto de hogar como de automóvil. La posibilidad de realizar una valoración a través de una imagen que nos permitirá ahorrar costes y desplazamientos de personal tanto para la empresa como para el usuario. Esto no significa que pueda eliminarse totalmente este servicio a domicilio, pero puede facilitar inicialmente siniestros que no revelen demasiada importancia.

3.- Igualmente, añadir el análisis de voz, puede suponer una ventaja para personas que por distintas capacidades, no puedan manipular objetos.

11. Referencias Bibliográficas

[1] R-statistics.com. (2018). ggplot2 | R-statistics blog. [online] Available at: https://www.r-statistics.com/tag/ggplot2/ [Accessed 2018].

[2]Stat.ethz.ch. (2018). [online] Available at: https://stat.ethz.ch/R-manual/R-devel/library/parallel/doc/parallel.pdf [Accessed 2018].

[3] Stat.ethz.ch. (2018). R: Parallel Versions of 'lapply' and 'mapply' using Forking. [online] Available at: https://stat.ethz.ch/R-manual/R-devel/library/parallel/html/mclapply.html [Accessed 2018].

[4]Cran.r-project.org. (2018). [online] Available at: https://cran.r-project.org/web/packages/ClusterR/ClusterR.pdf [Accessed 2018].

[5]Sthda.com. (2018). Types of Clustering Methods: Overview and Quick Start R Code - Articles - STHDA. [online] Available at: http://www.sthda.com/english/articles/25-cluster-analysis-in-r-practical-guide/111-types-of-clustering-methods-overview-and-quick-start-r-code/ [Accessed 13 Mar. 2018].

[6] Sthda.com. (2018). Profile of kassambara - STHDA. [online] Available at: http://www.sthda.com/english/user/profile/1 [Accessed 2018].

[7] Caminos aleatorios. (2018). Una introducción al paquete CARET. [online] Available at: https://caminosaleatorios.wordpress.com/2017/08/11/una-introduccion-al-paquete-caret/ [Accessed 8 Apr. 2018].

[8] Rochina, P., Rochina, P. and perfil, V. (2018). Python o R. ¿Qué lenguaje utilizar para el análisis de datos?. [online] Canal Informática y TICS. Available at: https://revistadigital.inesem.es/informatica-y-tics/python-r-analisis-datos/ [Accessed 8 Apr. 2018].

[9] CulturaCRM. (2018). KPI necesarios y sus características en Business Intelligence. [online] Available at: https://culturacrm.com/business-intelligence/kpi-business-intelligence/ [Accessed 8 Apr. 2018].

[10] Manuel, J. (2018). ¿Qué es un KPI en marketing?, definición, cómo hacerlo y ejemplos. [online] La Cultura del Marketing. Available at: https://laculturadelmarketing.com/que-es-un-kpi-en-marketing/ [Accessed 8 Apr. 2018].

[11] Docs.h2o.ai. (2018). Productionizing H2O — H2O 3.18.0.5 documentation. [online] Available at: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/productionizing.html [Accessed 7 Apr. 2018].

[12] Docs.h2o.ai. (2018). Gradient Boosting Machine (GBM) — H2O 3.18.0.5 documentation. [online] Available at: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/gbm.html [Accessed 7 Apr. 2018].

[13] Docs.h2o.ai. (2018). *Starting H2O — H2O 3.18.0.5 documentation*. [online] Available at: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/starting-h2o.html [Accessed 2018].

[14] h2o.gbm(), R. (2018). R: Plot trees from h2o.randomForest() and h2o.gbm(). [online] Stackoverflow.com. Available at: https://stackoverflow.com/questions/37017165/r-plot-trees-from-h2o-randomforest-and-h2o-gbm [Accessed 2018].

[15] Docs.h2o.ai. (2018). h2o-genmodel version 3.18.0.5 API. [online] Available at: http://docs.h2o.ai/h2o/latest-stable/h2o-genmodel/javadoc/index.html [Accessed 2018].

[16] Rpsychologist.com. (2018). Understanding and Interpreting Correlations - an Interactive Visualization. [online] Available at: http://rpsychologist.com/d3/correlation/ [Accessed 2018].

[17] Rpsychologist.com. (2018). Interpreting Confidence Intervals - an Interactive Visualization. [online] Available at: http://rpsychologist.com/d3/CI/ [Accessed 2018].

[18] Setosa.io. (2018). Setosa data visualization and visual explanations. [online] Available at: http://setosa.io/#/ [Accessed 8 Apr. 2018].

[19] Explained Visually. (2018). Explained Visually. [online] Available at: http://setosa.io/ev/ [Accessed 8 Apr. 2018].

[20] Setosa.io. (2018). CSV Fingerprints. [online] Available at: http://setosa.io/blog/2014/08/03/csv-fingerprints/ [Accessed 8 Apr. 2018].

[21] Graphviz.gitlab.io. (2018). Windows Packages. [online] Available at: https://graphviz.gitlab.io/_pages/Download/Download_windows.html [Accessed 2018].

[22]Redondo, F. and Redondo, F. (2018). Postman: gestiona y construye tus APIs rápidamente - Paradigma. [online] Paradigma. Available at: https://www.paradigmadigital.com/dev/postman-gestiona-construye-tus-apis-rapidamente/ [Accessed 8 Apr. 2018].

[23] Codegen, S. (2018). World's Most Popular API Framework | Swagger. [online] Swagger. Available at: https://swagger.io/ [Accessed 8 Apr. 2018].

[24] Azure.microsoft.com. (2018). Getting Started with Azure API Management REST API. [online] Available at: https://azure.microsoft.com/en-us/resources/videos/getting-started-with-azure-api-management-rest-api/ [Accessed 7 Apr. 2018].

[25] Docs.microsoft.com. (2018). Power BI tutorial for Azure Cosmos DB connector. [online] Available at: https://docs.microsoft.com/en-us/azure/cosmos-db/powerbi-visualize#flattening-and-transforming-json-documents [Accessed. 2018].

[26] Mclibre.org. (2018). Introducción a la programación con Pyhton. Bartolomé Sintes Marco. [online] Available at: http://www.mclibre.org/consultar/python/ [Accessed 2018].

[27] Docs.python.org.ar. (2018). Tutorial de Python (y Django!) en Español. [online] Available at: http://docs.python.org.ar/tutorial/ [Accessed 2018].

[28] CódigoFacilito. (2018). Curso de Python Básico Gratis. [online] Available at: https://codigofacilito.com/cursos/Python [Accessed 2018].

[29] principiantes, P. (2018). Python para principiantes. [online] Librosweb.es. Available at: https://librosweb.es/libro/python/ [Accessed 2018].

[30] Acodigo.blogspot.com.es. (2018). Python Leer y escribir datos en formato JSON. [online] Available at: http://acodigo.blogspot.com.es/2017/03/python-leer-y-escribir-datos-en-formato.html [Accessed 2018].

[31] Evilnapsis.com. (2018). Introduccion a Crear Aplicaciones en Android Studio – Evilnapsis. [online] Available at: http://evilnapsis.com/2017/01/25/introduccion-a-crear-aplicaciones-en-android-studio/ [Accessed 2018].

[32] Studio, C. (2018). Conoce Android Studio | Android Studio. [online] Developer.android.com. Available at: https://developer.android.com/studio/intro/index.html?hl=es-419 [Accessed 2018].

[33] Docs.microsoft.com. (2018). Introducción a Power BI Desktop - Power BI. [online] Available at: https://docs.microsoft.com/es-es/power-bi/desktop-getting-started [Accessed 2018].

[34] Docs.microsoft.com. (2018). Tutorial: Uso de los ejemplos de Power BI - Power BI. [online] Available at: https://docs.microsoft.com/es-es/power-bi/sample-tutorial-connect-to-the-samples [Accessed 2018].

[35] YouTube. (2018). Power BI Tutorial - aprende con ejemplos prácticos. [online] Available at: https://www.youtube.com/watch?v=yfG6M0AAXFQ [Accessed 2018].

ANEXOS

Anexo A. Código javascript página web

CODIGO JavaScript que genera la Web

```
if (jQuery.browser["mobile"] == false){
    $("#myVideo").append('<source src="img/bkgv.mp4" type="video/mp4">');
    $("#birthdate").prop('type', 'text');
    const picker = datepicker(document.querySelector('#birthdate'), {
        minDate: new Date(1800,1,1),
        maxDate: new Date(),
        formatter: function(el, date) {
            // This will display the date as `1/1/2017`.
            var options = {};
            options.timeZone = 'UTC';
            el.value = date.toLocaleDateString('es-ES');
        },

        onSelect: function(instance) {
            // Show which date was selected.
            var r=instance.dateSelected;
            //console.log(instance.dateSelected);
        },


        customMonths: ['Ene', 'Feb', 'Mar', 'Abr', 'May', 'Jun', 'Jul', 'Ago', 'Sep',
'Oct', 'Nov', 'Dic'],
        customDays: ['D', 'L', 'M', 'X', 'J', 'V', 'S'],
        overlayPlaceholder: 'Escriba el aÃ±o con 4 dÃgitos',
        overlayButton: 'Go!'
    });

}
var autocomplete;
FACTOR='{"C1.A" : "0.75", "C1.B" : "0.85", "C1.C" : "1.10", "C2.A" : "1.20", "C2.B" :
"1.35", "C2.C" : "1.5", "D1.A" : "0.70", "D1.B" : "0.85", "D1.C" : "1.30"}';
titulos = ["","TERCEROS", "REALE DOS (SOLO LUNAS)", "TERCEROS AMPLIADO","TODO RIESGO
FRANQUICIA 300 EUROS"];



$("#submit").on("click", function(e) {
    e.preventdefault;
          //console.log("click");
    matricula=$("#license").val();
    //matricula="3465BZB";
          dni=$("#dni").val();
    //dni="01037702B";
    birthdate=$("#birthdate").val();
    //birthdate="01/01/1968";
    address=$("#address").val();
    //address="Cll Juan Perez ZuÃ±iga 41 2-E Madrid";
    filled=true;
    if (matricula == "" || dni == "" || birthdate == "" || address == ""){
        filled = false;
    }
    if ($("#checky")[0].checked == false) {
        filled=false;
    }
    if (filled == true ) {
        $('.container').hide();
        $("#myVideo").first().attr('src','');
        $("#myVideo").hide();
        $("#topbar").append("<img src='img/logo-200.jpg'>");
        $("#leftbar").append("<div class='leftbar_item'><span style='font-
weight:bold'>MatrÃcula:</span><br>"+matricula+"</div>");
        $("#leftbar").append("<div class='leftbar_item'><span style='font-
weight:bold'>DNI:</span> <br>"+dni+"</div>");
        $("#leftbar").append("<div class='leftbar_item'><span style='font-
weight:bold'>Fecha de Nacimiento:</span> <br>"+birthdate+"</div>");
        $("#leftbar").append("<div class='leftbar_item'><span style='font-
weight:bold'>DirecciÃ³n:</span> <br>"+address+"</div>");
        $("#topbar").show();
```

```
        $("#leftbar").show();
        $("#rightbar").append("<img src='img/tach.gif'>");
        $("#rightbar").show();
        setTimeout(function(){
            $.when(querydni()).done(function(a1) {
                cliente(a1);
            });
        },500);
    }else{
        //console.log ("ERROR");
        $(".modal-body").append("<p>Falta llenar campos obligatorios</p>");
        $("#myModal").show();

    }
});


$(".close").on("click", function(e) {
    e.preventdefault;
    $(".modal-body").empty();
    $("#myModal").hide();
});

function querydni() {
    console.log ("querydni");
    queryurl="https://pilotosmartpricingweb.azurewebsites.net/querydni.php";
    return $.ajax({
        type: "post",
        url: queryurl,
        tryCount : 0,
        retryLimit : 3,
        dataType:"text",
        data: {dni:dni},
        success: function (response) {
            //console.log(response);
            return response;
        },
        error: function(e) {
            console.log(e.responseText);
            this.tryCount++;
            if (this.tryCount <= this.retryLimit) {
                //try again
                $.ajax(this);
                return;
            }
            return;
        }
    });
}

function cliente(response) {
    //console.log("funcion cliente");
    objresponse=JSON.parse(response);
    console.log("Query:")
    console.log(objresponse);
    cliente=objresponse[0]["CLIENTE_ACTUAL"];
    if (cliente == "S") {
        console.log("es cliente");
        $.when(probability()).done(function(a1){
            posneg(a1);
        });
    }else{
        console.log("No es cliente");
        $.when(nocliente()).done(function(a1){
            scoring(a1);
        });
    }
}


function probability () {
    queryurl="https://productobigdataml.azurewebsites.net/probability";
```

```
    //json=' {"CANTIDAD_POLIZAS" : "9", "YEARS_VIGENTE1" : "20", "RAMOS": "2", "NOTA1":
"OTROS", "TOTAL_SINIES_DESDE2005": "2","COD_SEGMENTO_1": "999", "SOCRENTAMEDIA":
"1655.65", "DENSIDAD": "10049.78", "HABITANTES": "91896", "CLASIFICACION_EMPLEADO": "NO
EMPLEADO"}';
    var objProb = {};
    objProb["CANTIDAD_POLIZAS"] = objresponse[0]["CANTIDAD_POLIZAS"];
    objProb["YEARS_VIGENTE1"] = objresponse[0]["YEARS_VIGENTE"];
    objProb["RAMOS"] =  objresponse[0]["RAMOS"];
    objProb["NOTA1"] =  objresponse[0]["NOTA1"];
    objProb["TOTAL_SINIES_DESDE2005"] =  objresponse[0]["TOTAL_SINIES_DESDE2005"];
    objProb["COD_SEGMENTO_1"] =  objresponse[0]["COD_SEGMENTO_1"];
    objProb["SOCRENTAMEDIA"] =  objresponse[0]["SOCRENTAMEDIA"];
    objProb["DENSIDAD"] = objresponse[0]["DENSIDAD"];
    objProb["HABITANTES"] =  objresponse[0]["HABITANTES"];
    objProb["CLASIFICACION_EMPLEADO"] =  objresponse[0]["CLASIFICACION_EMPLEADO"];
    jsonProb=JSON.stringify(objProb);
    //console.log(objProb);
    return $.ajax({
        type: "post",
        url: queryurl,
        dataType:"text",
        tryCount : 0,
        retryLimit : 3,
        data: jsonProb,
        contentType: "application/json",
        success: function (response2) {
            console.log("Probability:"+response2);
        },
        error: function(e) {
            console.log(e.responseText);
            this.tryCount++;
            if (this.tryCount <= this.retryLimit) {
                //try again
                $.ajax(this);
                return;
            }
            return;
        }
    });
}

function posneg (response2) {
    //console.log("posneg "+ response2);
    objProbResponse=JSON.parse(response2);
    //console.log(objProbResponse);
    probValue=objProbResponse["message"]
    if (probValue == 0) {
        console.log("negative");
        $.when(negative()).done(function(a1){
            scoring(a1);
        });
    }else{
        console.log("positive");
        $.when(positive()).done(function(a1){
            scoring(a1);
        });
    }
}

function negative () {
    queryurl="https://productobigdataml.azurewebsites.net/negative";
    json='{"TOTAL_SINIES_DESDE2005": "0", "CANTIDAD_POLIZAS": "1", "COD_SEGMENTO_1":
"100", "COD_PROVINCIA": "28", "YEARS_VIGENTE1": "1", "SOCGRUPOMOSAIC": "J ", "RAMOS": "1",
"DIAS_TMAX_30": "100", "SOCPAROTOT": "100", "NOTA2V": "1", "EDAD": "36", "NOTA1": "OTROS",
"SOCSEGMENTOSEGUROS": "TA2", "R_MAX_VEL": "100", "TMIN": "100", "DTORMENTA": "100",
"DP100": "100", "DNIEBLA": "100"}';
    var objNeg = {};
    objNeg["TOTAL_SINIES_DESDE2005"]= objresponse[0]["TOTAL_SINIES_DESDE2005"];
    objNeg["CANTIDAD_POLIZAS"]= objresponse[0]["CANTIDAD_POLIZAS"];
    objNeg["COD_SEGMENTO_1"]= objresponse[0]["COD_SEGMENTO_1"];
    objNeg["COD_PROVINCIA"]= objresponse[0]["COD_PROVINCIA"];
    objNeg["YEARS_VIGENTE1"]= objresponse[0]["YEARS_VIGENTE"];
    objNeg["SOCGRUPOMOSAIC"]= objresponse[0]["SOCGRUPOMOSAIC"];
```

```
    objNeg["RAMOS"]= objresponse[0]["RAMOS"];
    objNeg["DIAS_TMAX_30"]= objresponse[0]["DIAS_TMAX_30"];
    objNeg["SOCPAROTOT"]= objresponse[0]["SOCPAROTOT"];
    objNeg["NOTA2V"]= objresponse[0]["NOTA2V"];
    objNeg["EDAD"]= objresponse[0]["EDAD"];
    objNeg["NOTA1"]= objresponse[0]["NOTA1"];
    objNeg["SOCSEGMENTOSEGUROS"]= objresponse[0]["SOCSEGMENTOSEGUROS"];
    objNeg["R_MAX_VEL"]= objresponse[0]["R_MAX_VEL"];
    objNeg["TMIN"]= objresponse[0]["TMIN"];
    objNeg["DTORMENTA"]= objresponse[0]["DTORMENTA"];
    objNeg["DP100"]= objresponse[0]["DP100"];
    objNeg["DNIEBLA"]= objresponse[0]["DNIEBLA"];
    jsonNeg=JSON.stringify(objNeg);
    //console.log(objNeg);
    return $.ajax({
        type: "post",
        url: queryurl,
        tryCount : 0,
        retryLimit : 3,
        dataType:"text",
        data: jsonNeg,
        contentType: "application/json",
        success: function (response3) {
            console.log("negative:"+response3);
        },
        error: function(e) {
            console.log(e.responseText);
            this.tryCount++;
            if (this.tryCount <= this.retryLimit) {
                //try again
                $.ajax(this);
                return;
            }
            return;
        }
    });
}

function positive () {
    queryurl="https://productobigdataml.azurewebsites.net/positive";
    json='{ "COD_SEGMENTO_1": "100","CANTIDAD_POLIZAS": "2","YEARS_VIGENTE1":
"3","TOTAL_SINIES_DESDE2005": "2","COD_PROVINCIA": "28","NOTA1": "OTROS","NOTA2V":
"1","RAMOS": "1","CONT_VIGOR": "1","SOCGRUPOMOSAIC": "J ","SOCSEGMENTOSEGUROS":
"TA2","EDAD": "34"}';
    var objPos = {};
    objPos["COD_SEGMENTO_1"]= objresponse[0]["COD_SEGMENTO_1"];
    objPos["CANTIDAD_POLIZAS"]= objresponse[0]["CANTIDAD_POLIZAS"];
    objPos["YEARS_VIGENTE1"]= objresponse[0]["YEARS_VIGENTE1"];
    objPos["TOTAL_SINIES_DESDE2005"]=objresponse[0]["TOTAL_SINIES_DESDE2005"];
    objPos["COD_PROVINCIA"]= objresponse[0]["COD_PROVINCIA"];
    objPos["NOTA1"]= objresponse[0]["NOTA1"];
    objPos["NOTA2V"]= objresponse[0]["NOTA2V"];
    objPos["RAMOS"]= objresponse[0]["RAMOS"];
    objPos["CONT_VIGOR"]= objresponse[0]["CONT_VIGOR"];
    objPos["SOCGRUPOMOSAIC"]= objresponse[0]["SOCGRUPOMOSAIC"];
    objPos["SOCSEGMENTOSEGUROS"]= objresponse[0]["SOCSEGMENTOSEGUROS"];
    objPos["EDAD"]= objresponse[0]["EDAD"];
    jsonPos=JSON.stringify(objPos);
    //console.log(objPos);
    return $.ajax({
        type: "post",
        url: queryurl,
        dataType:"text",
        tryCount : 0,
        retryLimit : 3,
        data: jsonPos,
        contentType: "application/json",
        success: function (response3) {
            console.log("positive:"+response3);
        },
        error: function(e) {
            console.log(e.responseText);
            this.tryCount++;
```

```
            if (this.tryCount <= this.retryLimit) {
                //try again
                $.ajax(this);
                return;
            }
            return;
        }
    });
}

function nocliente () {
    queryurl="https://productobigdataml.azurewebsites.net/noclient";
    json= '{"NOTA1": "OTROS", "NOTA2V": "1", "COD_PROVINCIA": "28", "EDAD": "40",
"SOCGRUPOMOSAIC": "J ", "SOCPAROTOT": "100", "SOCURBANIDAD": "100", "SOCSEGMENTOSEGUROS":
"TA2", "SOCRENTAMEDIA": "100"}';
    var objNoc = {};
    objNoc["NOTA1"]= objresponse[0]["NOTA1"];
    objNoc["NOTA2V"]= objresponse[0]["NOTA2V"];
    objNoc["COD_PROVINCIA"]= objresponse[0]["COD_PROVINCIA"];
    objNoc["EDAD"]= objresponse[0]["EDAD"];
    objNoc["SOCGRUPOMOSAIC"]= objresponse[0]["SOCGRUPOMOSAIC"];
    objNoc["SOCPAROTOT"]= objresponse[0]["SOCPAROTOT"];
    objNoc["SOCURBANIDAD"]= objresponse[0]["SOCURBANIDAD"];
    objNoc["SOCSEGMENTOSEGUROS"]= objresponse[0]["SOCSEGMENTOSEGUROS"];
    objNoc["SOCRENTAMEDIA"]= objresponse[0]["SOCRENTAMEDIA"];
    jsonNoc=JSON.stringify(objNoc);
    //console.log(objNoc);
    return $.ajax({
        type: "post",
        url: queryurl,
        dataType:"text",
        tryCount : 0,
        retryLimit : 3,
        data: jsonNoc,
        contentType: "application/json",
        success: function (response3) {
            console.log("No Cliente:"+response3);
        },
        error: function(e) {
            console.log(e.responseText);
            this.tryCount++;
            if (this.tryCount <= this.retryLimit) {
                //try again
                $.ajax(this);
                return;
            }
            return;
        }
    });
}

function scoring (response4) {
    objmargin=JSON.parse(response4);
    objfactor=JSON.parse(FACTOR);
    margin=objmargin["message"]
    console.log("margin:"+margin);
    tarifa2=objresponse[0]["TARIFA_3"];
    scoring=(margin/tarifa2)*100;
    console.log("scoring:"+scoring);
    if (cliente == "S") {
        if (scoring >= 0){
            if (scoring >= 60) {
                console.log("C1.A");
                factor=objfactor["C1.A"];
            }
            if (scoring < 60 && scoring >=25){
                console.log("C1.B");
                factor=objfactor["C1.B"];
            }
            if (scoring < 25) {
                console.log("C1.C");
                factor = objfactor["C1.C"];
            }
```

```
        }else{
            if (scoring >= -36){
                console.log("C2.A");
                factor=objfactor["C2.A"];
            }
            if (scoring <-36 && scoring >= -60) {
                console.log("C2.B");
                factor=objfactor["C2.B"];
            }
            if  (scoring < -60) {
                console.log("C2.C");
                factor=objfactor["C2.C"];
            }
        }
    }else{
        if (scoring >= 50) {
            console.log("D1.A");
            factor=objfactor["D1.A"];
        }
        if (scoring < 50 && scoring >=15) {
            console.log("D1.B");
            factor=objfactor["D1.B"];
        }
        if (scoring < 15) {
            console.log("D1.C");
            factor=objfactor["D1.C"];
        }
    }
    console.log("factor:"+factor);
    $("#rightbar").empty();
    $("#rightbar").append('<div id="presupuesto">SU PRESUPUESTO</div>');

    for (i = 1; i < 5; i++) {
        methodName="TARIFA_"+i;
        tarifaclass="tarifaL";
        /*
        line=i%2;
        if (line != 0){
            tarifaclass=tarifaclass+"L";
        }else{
            tarifaclass=tarifaclass+"R";
        }
        */
        newvalue=Math.floor(objresponse[0][methodName]*factor);
        var tarifatext='<div class="'+tarifaclass+'">';
        var titulo = '<div class="tarifatitulo">'+titulos[i]+'</div>';
        var tarifaold='<div class="tarifaold">'+objresponse[0][methodName]+' â,¬</div>';
        var tarifanew='<div class="tarifanew">'+newvalue+' â,¬</div>';
        var tarifaanual='<div class="tarifaanual">TOTAL ANUAL</div>';
        $("#rightbar").append(tarifatext+titulo+tarifaold+tarifanew+tarifaanual+'</div>');
    }
    return "FIN";
}

function initAutocomplete() {
    // Create the autocomplete object, restricting the search to geographical
    // location types.
    autocomplete = new google.maps.places.Autocomplete(
        /** @type {!HTMLInputElement} */(document.getElementById('address')),
        {types: ['geocode']});

    // When the user selects an address from the dropdown, populate the address
    // fields in the form.
    autocomplete.addListener('place_changed', fillInAddress);
}

function fillInAddress() {
    // Get the place details from the autocomplete object.
    var place = autocomplete.getPlace();
    /*
    for (var component in componentForm) {
      document.getElementById(component).value = '';
      document.getElementById(component).disabled = false;
```

```
    }
    */
}

function geolocate() {
    /*
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(function(position) {
        var geolocation = {
            lat: position.coords.latitude,
            lng: position.coords.longitude
        };
        var circle = new google.maps.Circle({
            center: geolocation,
            radius: position.coords.accuracy
        });
        autocomplete.setBounds(circle.getBounds());
        });
    }
    */
}
```

Anexo B. Parámetros de entrada y salida del webservice del catastro.

Parámetros de entrada.

Provincia: Obligatorio. Denominación de una provincia según lo devuelto en el listado de provincias.
Municipio: Obligatorio. Denominación de un municipio según lo devuelto en el listado de municipios.
TipoVia: Obligatorio. Abreviatura del tipo de vía. Ver listado de abreviaturas en el Anexo I.
NombreVía: Obligatorio. Cadena con el nombre o parte del nombre de la vía.
Número: Obligatorio. Número del que se desea conocer la referencia catastral.
Bloque: Opcional.
Escalera: Opcional
Planta: Opcional
Puerta: Opcional

Formato de salida

Como se ha indicado anteriormente, el servicio puede devolver:
1.- Una lista con candidatos en caso de que la provincia, municipio, vía o número no existan. El formato de salida es el indicado en los puntos anteriores.
2.- Una lista de todos los inmuebles que coinciden con los criterios de búsqueda, en cuyo caso el formato de salida es:

```
<consulta_dnp>
<control>
<cudnp>NÚMERO DE ITEMS EN LA LISTA DE BIENES INMUEBLES</cudnp>
</control>
<lrcdnp> LISTA DE BIENES INMUEBLES
<rcdnp> DATOS DE UN INMUEBLE
<rc>
<pc1> POSICIONES 1-7 DE LA REFERENCIA CATASTRAL (RC) DEL INMUEBLE</pc1>
<pc2>POSICIONES 8-14 DE LA RC DEL INMUEBLE</pc1>
<car>POSICIONES 15-19 DE LA RC (CARGO)</car>
<cc1>PRIMER DÍGITO DE CONTROL DE LA RC</cc1>
<cc2>SEGUNDO DÍGITO DE CONTROL DE LA RC </cc2>
</rc>
<dt>DOMICILIO TRIBUTARIO DEL INMUEBLE
<lourb>LOCALIZACIÓN DE INMUEBLE URBANO
<loint>
<bq>BLOQUE</bq>
<es>ESCALERA</es>
<pt>PLANTA</pt>
<pu>PUERTA</pu>
</loint>
</lourb>
<lorus>
<cma>CÓDIGO DEL MUNICIPIO AGREGADO</cma>
<czc>CÓDIGO DE LA ZONA DE CONCENTRACIÓN</czc>
<cpp>
<cpo>CÓDIGO DEL POLÍGONO</cpo>
<cpa>CÓDIGO DE LA PARCELA</cpa>
</cpp>
<npa>NOMBRE DEL PARAJE</npa>
<cpaj>CÓDIGO DEL PARAJE</cpaj>
<lorus/>
</dt>
</rcdnp>
...
</lrcdnp>
</consulta_dnp>
```

3.- Los datos no protegidos de un inmueble, en cuyo caso el formato de salida es:

```
<consulta_dnp>
<control>
<cudnp>NÚMERO DE INMUEBLES DE LOS QUE SE PROPORCIONAN DATOS</cudnp>
<cucons>NÚMERO DE UNIDADES CONSTRUCTIVAS (INCLUYENDO ELEMENTOS COMUNES)</cucons>
<cucul>NUMERO DE SUBPARCELAS (CULTIVOS)</cucul>
</control>
<bico>
<bi>
<idbi>
<cn>TIPO DE BIEN INMUEBLE</cn>
<rc>
<pc1> POSICIONES 1-7 DE LA REFERENCIA CATASTRAL (RC) DEL INMUEBLE</pc1>
<pc2>POSICIONES 8-14 DE LA RC DEL INMUEBLE</pc1>
<car>POSICIONES 15-19 DE LA RC (CARGO)</car>
<cc1>PRIMER DÍGITO DE CONTROL DE LA RC</cc1>
<cc2>SEGUNDO DÍGITO DE CONTROL DE LA RC </cc2>
</rc>
</idbi>
<ldt>DOMICILIO TRIBUTARIO NO ESTRUCTURADO (TEXTO)</ldt>
<debi> DATOS ECONÓMICOS DEL INMUEBLE
<luso>Residencial</luso>
<sfc>SUPERFICIE</sfc>
<cpt>COEFICIENTE DE PARTICIPACIÓN</cpt>
<ant>ANTIGUEDAD</ant>
</debi>
</bi>
<lcons>LISTA DE UNIDADES CONSTRUCTIVAS
<cons>UNIDAD CONSTRUCTIVA
<lcd>USO DE LA UNIDAD CONSTRUCTIVA</lcd>
<dt>
<lourb>
<loint>
<es>ESCALERA</es>
<pt>PLANTA</pt>
<pu>PUERTA</pu>
</loint>
</lourb>
</dt>
<dfcons>
<stl>SUPERFICIE DE LA UNIDAD CONSTRUCTIVA</stl>
</dfcons>
</cons>
<cons>
<dfcons>
<stl>SUPERFICIE DE LOS ELEMENTOS COMUNES</stl>
</dfcons>
</cons>
</lcons>
<lspr>LISTA DE SUBPARCELAS
<spr>SUBPARCELA
<cspr>CÓDIGO DE SUBPARCELA</cspr>
<dspr>DATOS DE SUBPARCELA
<ccc>CALIFICACIÓN CATASTRAL</ccc>
<dcc>DENOMINACIÓN DE LA CLASE CULTIVO</dcc>
<ip>INTENSIDAD PRODUCTIVA</ip>
<ssp>SUPERFICIE DE LA SUBPARCELA EN METROS CUADRADOS</ssp>
</dspr>
</spr>
</lspr>
</bico>
</consulta_dnp>
```

Anexo C. Código Python

```
def modelo():
        dni       = str(request.args.get('dni'))
        matricula = request.args.get('matricula')
        fecha_nacimiento = request.args.get('fecha_nacimiento')
        direccion = request.args.get('direccion')
        uri       = "mongodb://rsgdsmongodb:gw1kdadbuISj2TjK0Wt6nKTnuuPk4w8EWPDwU0ixtQqM3c2d8DzGCbxIjo7Aw1V7jRknPn3fhw28i
        client    = MongoClient(uri)
        db        = client.smart_pricing
        col       = db.variables_modelos_ml
        _id       = col.find_one({"DNI":dni})
        if _id is None:
           _id       = col.find_one()

        cliente_actual = _id["CLIENTE_ACTUAL"]
        idcliente = _id["IDCLIENTE"]
        #dni = _id["DNI"]
        cont_vigor = _id["CONT_VIGOR"]
        cod_provincia = _id["COD_PROVINCIA"]
        provincia = _id["PROVINCIA"]
        clasificacion_empleo = _id["CLASIFICACION_EMPLEADO"]
        nota1 = _id["NOTA1"]
        nota2v =_id["NOTA2V"]
        socsegmentoseguros = _id["SOCSEGMENTOSEGUROS"]
        socgrupomosaic = _id["SOCGRUPOMOSAIC"]
        cod_segmento = _id["COD_SEGMENTO"]
        cod_segmento_1 = _id["COD_SEGMENTO_1"]
        years_vigente = _id["YEARS_VIGENTE"]
        edad = _id["EDAD"]
        socrentamedia = _id["SOCRENTAMEDIA"]
        socstatus = _id["SOCSTATUS"]
        socvidmuerte = _id["SOCVIDMUERTE"]
        socvidmuervenc = _id["SOCVIDMUERVENC"]
        sochipot = _id["SOCHIPOT"]
        soccontcia = _id["SOCCONTCIA"]
        soccontmediador = _id["SOCCONTMEDIADOR"]
        socseguromedico  = _id["SOCSEGUROMEDICO"]
        socsegmedmarc = _id["SOCSEGMEDPARC"]
        socasegpagasegmed = _id["SOCASEGPAGASEGMED"]
        socsegmedemp = _id["SOCSEGMEDEMP"]
        socmotprecio = _id["SOCMOTPRECIO"]
        socmotofer = _id["SOCMOTOFER"]
        socmotconf = _id["SOCMOTCONF"]
        socmotnece = _id["SOCMOTNECE"]
        socmotserv = _id["SOCMOTSERV"]
        socter = _id["SOCTER"]
        socteropc = _id["SOCTEROPC"]
        soctrcf = _id["SOCTRCF"]
        soctrsf = _id["SOCTRSF"]
```

```
soccontautcia = _id["SOCCONTAUTCIA"]
soccontautmediador = _id["SOCCONTAUTMEDIADOR"]
soccontautbanco = _id["SOCCONTAUTBANCO"]
socmotrapi = _id["SOCMOTRAPI"]
soccontauttelf = _id["SOCCONTAUTTELF"]
soccontautint = _id["SOCCONTAUTINT"]
soccontautpers = _id["SOCCONTAUTPERS"]
socsiemprecia = _id["SOCSIEMPRECIA"]
socantma5 = _id["SOCANTMA5"]
soccoches1 = _id["SOCCOCHES1"]
soccoches0 = _id["SOCCOCHES0"]
socestres = _id["SOCESTRES"]
socagotamiento = _id["SOCAGOTAMIENTO"]
socinseghogar = _id["SOCSINSEGHOGAR"]
soccontm5 = _id["SOCCONTM5"]
soccomprainternet = _id["SOCCOMPRAINTERNET"]
soctipologia = _id["SOCTIPOLOGIA"]
socffamilias = _id["SOCFFAMILIAS"]
soctipviv = _id["SOCTIPVIV"]
socurbanidad = _id["SOCURBANIDAD"]
soctransitoriedad = _id["SOCTRANSITORIEDAD"]
soccalidadviv = _id["SOCCALIDADVIV"]
socfhogar = _id["SOCFHOGAR"]
tmin = _id["TMIN"]
dias_tmin_5 = _id["DIAS_TMIN_5"]
dias_tmas_30 = _id["DIAS_TMAX_30"]
dp100 = _id["DP100"]
dnieve = _id["DNIEVE"]
dtormenta = _id["DTORMENTA"]
dniebla = _id["DNIEBLA"]
drocio = _id["DROCIO"]
r_max_vel = _id["R_MAX_VEL"]
densidad = _id["DENSIDAD"]
ext_ext = _id["EXT_EXT"]
habitantes = _id["HABITANTES"]
socparotot = _id["SOCPAROTOT"]
seccioncensal = _id["SECCIONCENSAL"]
codigo_postal = _id["CODIGO_POSTAL"]
cantidad_polizas = _id["CANTIDAD_POLIZAS"]
prima_bruta = _id["PRIMA_BRUTA"]
margen_bruto = _id["MARGEN_BRUTO"]
total_sinies_desde2005 = _id["TOTAL_SINIES_DESDE2005"]
ramos = _id["RAMOS"]
tarifa_1 = _id["TARIFA_1"]
tarifa_2 = _id["TARIFA_2"]
tarifa_3 = _id["TARIFA_3"]
tarifa_4 = _id["TARIFA_4"]
```

```
        return bigdataml(
        cliente_actual ,
        idcliente ,
        dni ,
        cont_vigor ,
        cod_provincia ,
        provincia ,
        clasificacion_empleo ,
        nota1 ,
        nota2v ,
        socsegmentoseguros ,
        socgrupomosaic ,
        cod_segmento ,
        cod_segmento_1 ,
        years_vigente ,
        edad ,
        socrentamedia ,
        socstatus ,
        socvidmuerte ,
        socvidmuervenc ,
        sochipot ,
        soccontcia ,
        soccontmediador ,
        socseguromedico  ,
        socsegmedmarc ,
        socasegpagasegmed ,
        socsegmedemp ,
        socmotprecio ,
        socmotofer ,
        socmotconf ,
        socmotnece ,
        socmotserv ,
        socter ,
        socteropc ,
        soctrcf ,
        soctrsf ,
        soccontautcia ,
        soccontautmediador ,
        soccontautbanco ,
        socmotrapi ,
        soccontauttelf ,
        soccontautint ,
        soccontautpers ,
        socsiemprecia ,
        socantma5 ,
        soccoches1 ,
        soccoches0 ,
        socestres ,
        socagotamiento ,
        socinseghogar ,
        soccontm5 ,
        soccomprainternet ,
        soctipologia ,
        socffamilias ,
        soctipviv ,
        socurbanidad ,
        soctransitoriedad ,
        soccalidadviv ,
        socfhogar ,
        tmin ,
        dias_tmin_5 ,
        dias_tmas_30 ,
        dp100 ,
        dnieve ,
        dtormenta ,
        dniebla ,
        drocio ,
        r_max_vel ,
        densidad ,
        ext_ext ,
        habitantes ,
        socparotot ,
        seccioncensal ,
        codigo_postal ,
        cantidad_polizas ,
        prima_bruta ,
        margen_bruto ,
        total_sinies_desde2005 ,
        ramos ,
        tarifa_1 ,
        tarifa_2 ,
        tarifa_3 ,
        tarifa_4,
        matricula,
        direccion,
        fecha_nacimiento)
        client.close()
```

Anexo D. Funcionalidad de bigdatahtml

```
def bigdataml(
        cliente_actual ,
        idcliente ,
        dni ,
        cont_vigor ,
        cod_provincia ,
        provincia ,
        clasificacion_empleo ,
        nota1 ,
        nota2v ,
        socsegmentoseguros ,
        socgrupomosaic ,
        cod_segmento ,
        cod_segmento_1 ,
        years_vigente ,
        edad ,
        socrentamedia ,
        socstatus ,
        socvidmuerte ,
        socvidmuervenc ,
        sochipot ,
        soccontcia ,
        soccontmediador ,
        socseguromedico  ,
        socsegmedmarc ,
        socasegpagasegmed ,
        socsegmedemp ,
        socmotprecio ,
        socmotofer ,
        socmotconf ,
        socmotnece ,
        socmotserv ,
        socter ,
        socteropc ,
        soctrcf ,
        soctrsf ,
        soccontautcia ,
        soccontautmediador ,
        soccontautbanco ,
        socmotrapi ,
        soccontauttelf ,
        soccontautint ,
        soccontautpers ,
        socsiemprecia ,
        socantma5 ,
        soccoches1 ,
        soccoches0 ,
        socestres ,
        socagotamiento ,
        socinseghogar ,
        soccontm5 ,
        soccomprainternet ,
        soctipologia ,
        socffamilias ,
        soctipviv ,
        socurbanidad ,
        soctransitoriedad ,
        soccalidadviv ,
        socfhogar ,
        tmin ,
        dias_tmin_5 ,
        dias_tmas_30 ,
        dp100 ,
        dnieve ,
        dtormenta ,
        dniebla ,
        drocio ,
        r_max_vel ,
        densidad ,
        ext_ext ,
        habitantes ,
        socparotot ,
        seccioncensal ,
```

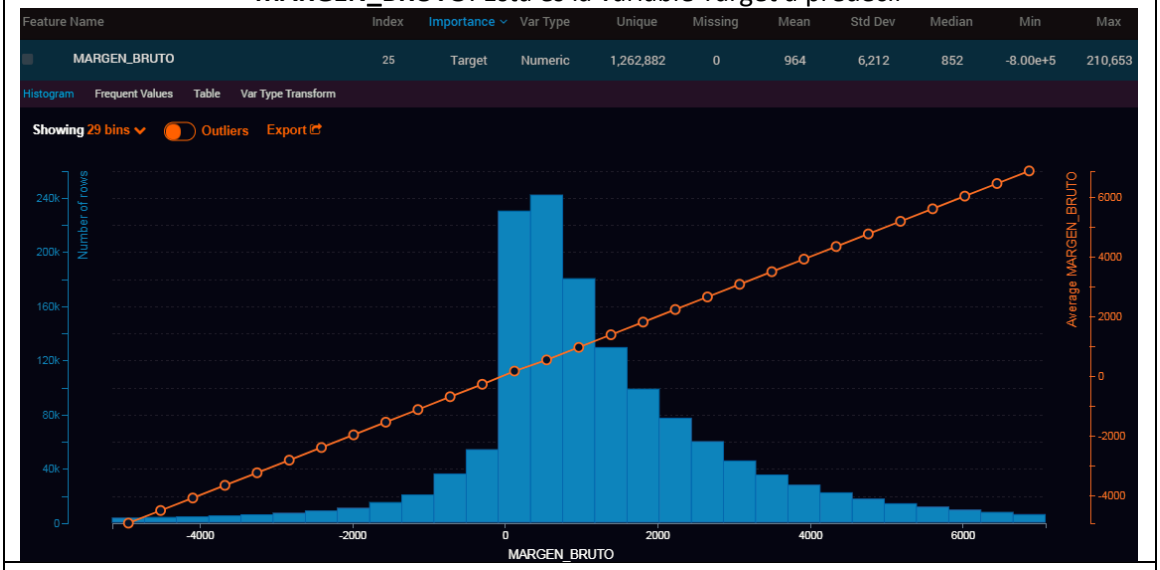
```
        codigo_postal ,
        cantidad_polizas ,
        prima_bruta ,
        margen_bruto ,
        total_sinies_desde2005 ,
        ramos ,
        tarifa_1 ,
        tarifa_2 ,
        tarifa_3 ,
        tarifa_4,
        matricula,
        direccion,
        fecha_nacimiento):
        if cliente(cliente_actual) == '1':
            aux_cliente = 1
            if probablity(cantidad_polizas ,
               years_vigente ,
               ramos ,
               nota1 ,
               total_sinies_desde2005 ,
               cod_segmento ,
               socrentamedia ,
               densidad ,
               abitantes ,
               clasificacion_empleado ,
               socsegmentoseguros) == '1':
                 resultado = positivos(cod_segmento_1 ,
                             cantidad_polizas ,
                             years_vigente1 ,
                             total_sinies_desde2005 ,
                             cod_provincia ,
                             nota1 ,
                             nota2v ,
                             ramos ,
                             cont_vigor ,
                             socgrupomosaic ,
                             socsegmentoseguros ,
                             edad)
            else:
                resultado = negativo(total_sinies_desde2005 ,
                            cantidad_polizas ,
                            cod_segmento ,
                            cod_provincia ,
                            years_vigente ,
                            socgrupomosaic ,
                            ramos ,
                            dias_tmax_30 ,
                            socparotot ,
                            nota2v ,
                            edad ,
                            nota1 ,
                            socsegmentoseguros ,
                            r_max_vel ,
                            tmin ,
                            dtormenta ,
                            dp100 ,
                            dniebla)
        else:
            resultado = noclientes(nota1 ,
                        nota2v ,
                        cod_provincia ,
                        edad ,
                        socgrupomosaic ,
                        socparotot ,
                        socurbanidad ,
                        socsegmentoseguros ,
                        socrentamedia)
            aux_cliente = 0
        return str(scoring(resultado,
                       tarifa_1,
                       tarifa_2,
                       tarifa_3,
                       tarifa_4,
                       aux_cliente,
                       dni,
                       matricula,
                       direccion,
                       fecha_nacimiento,
                       cod_provincia))
```

Anexo E. VARIABLES UTILIZADAS EN LOS MODELOS MACHINE LEARNING

VARIABLES UTILIZADAS EN LOS MODELOS

En cada una de las gráficas se incluye el margen bruto medio como segunda variable a graficar.

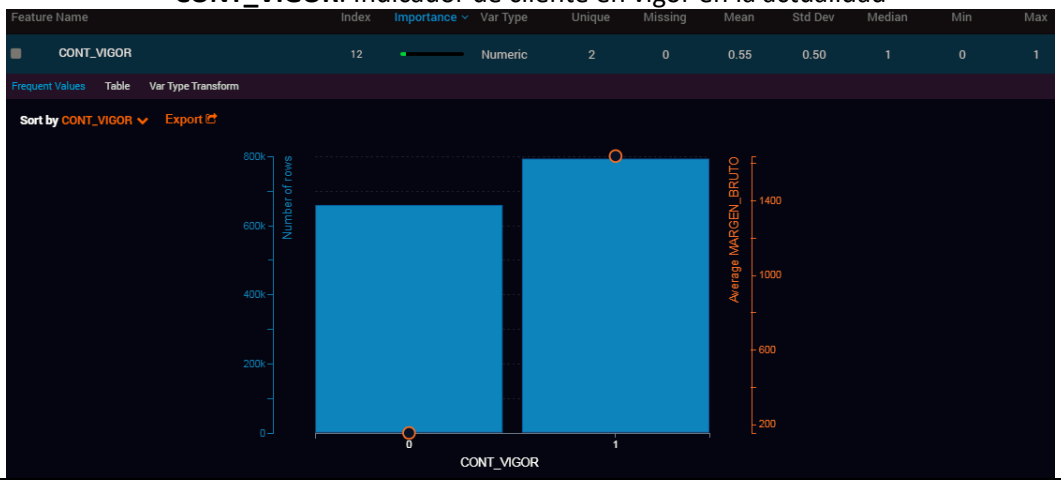
MARGEN_BRUTO: Está es la variable Target a predecir



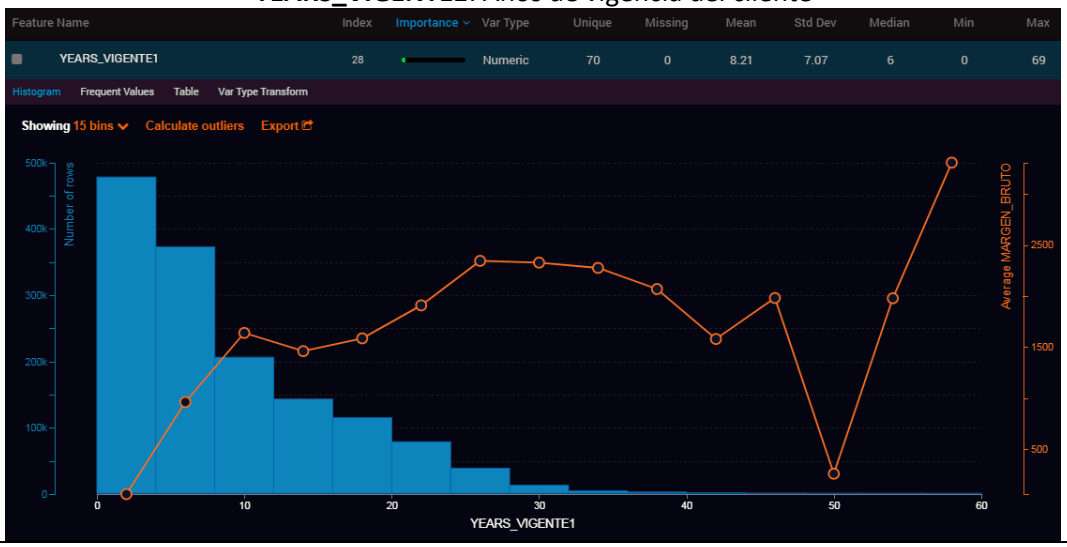
PRIMA_BRUTA:

TOTAL_SINIES_DESDE2005: Cantidad de siniestros

CONT_VIGOR: Indicador de cliente en vigor en la actualidad



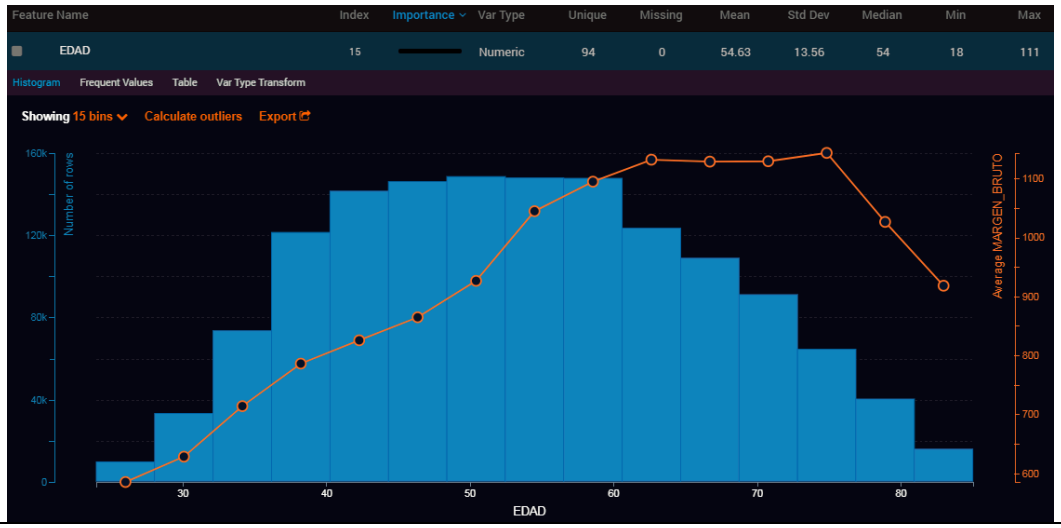
YEARS_VIGENTE1: Años de vigencia del cliente



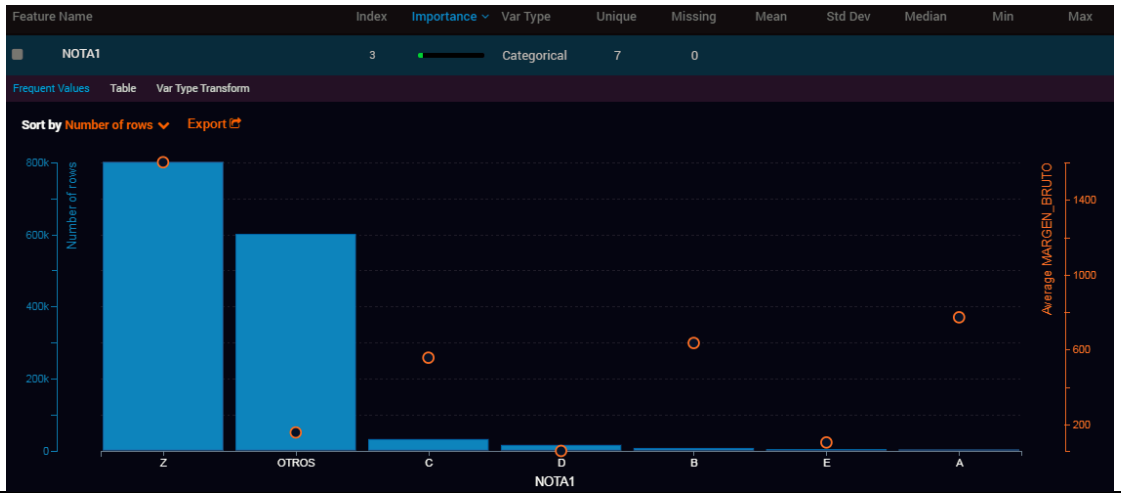
CANTIDAD_POLIZAS: Número de pólizas que tiene contratadas el cliente

RAMOS: Cantidad de Ramos diferentes que tiene contratados (Auto, Hogar, Diversos, etc)

EDAD:



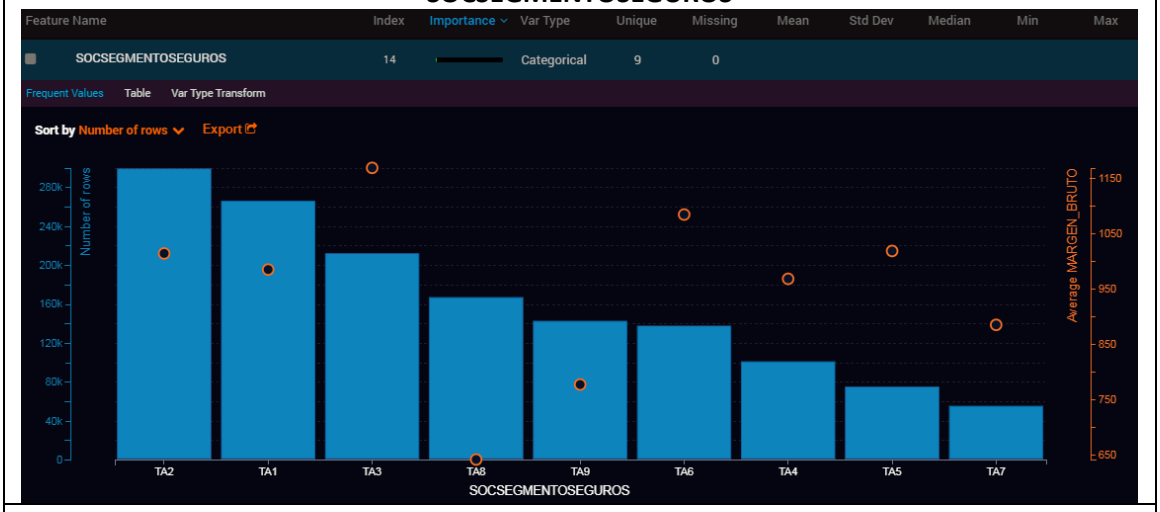
NOTA1: Indicador asociado a la morosidad del cliente



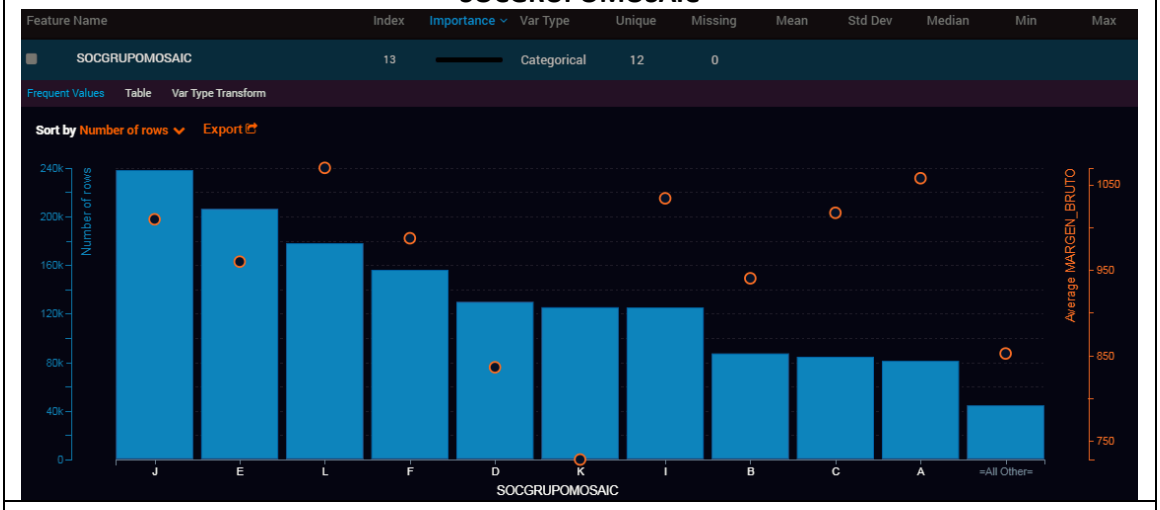
NOT2V:

COD_SEGMENTO1: Clasificación por segmentos definidos por Reale

SOCSEGMENTOSEGUROS



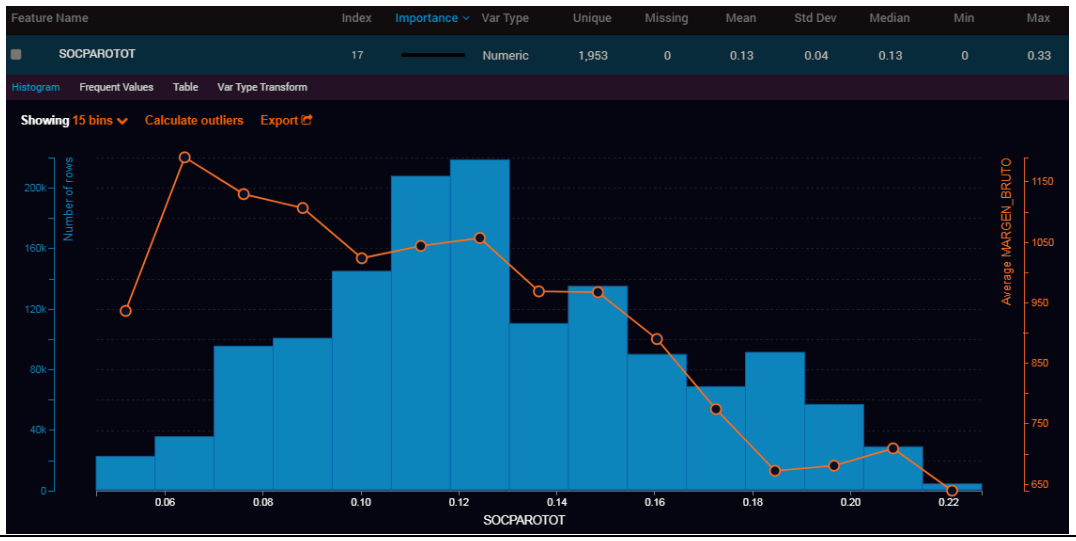
SOCGRUPOMOSAIC



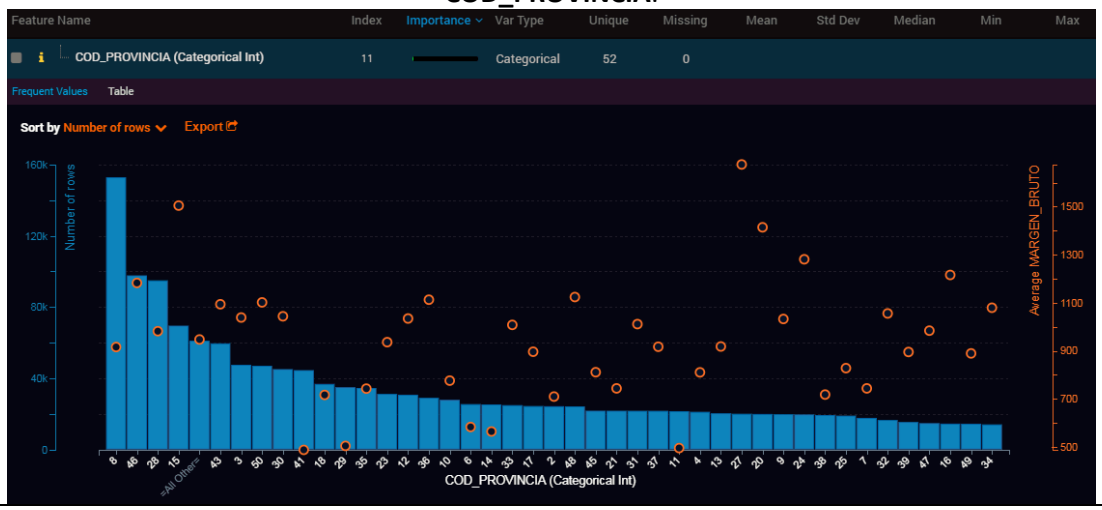
SOCRENTAMEDIA

SOCURBANIDAD

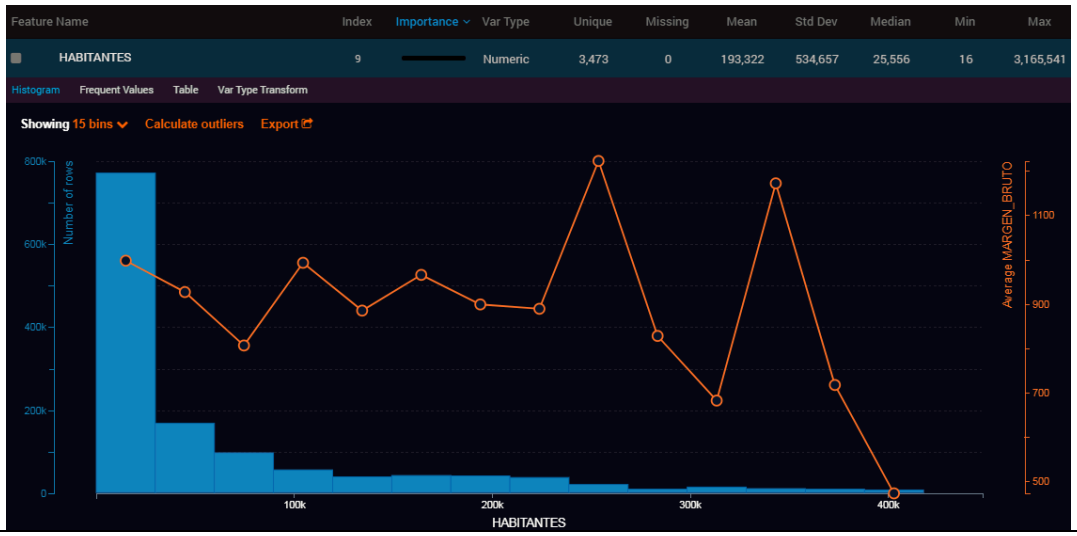
SOCPAROTOT



COD_PROVINCIA:

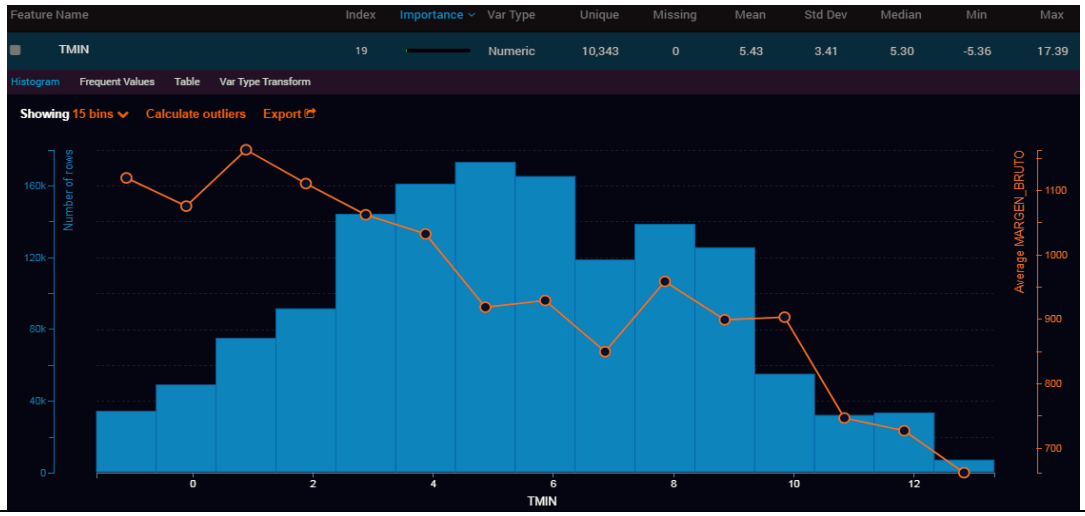


HABITANTES:



DENSIDAD:

TMIN



R_MAX_VEL:

DNIEBLA

DTORMENTA:

DIAS_TMAX_30

DP100

CLASIFICACION_EMPLEADO

Anexo F. PARAMETROS MODELO NOCLIENT

Parámetros:

Parameter	Value	Description
model_id	NOCLIENT	Destination id for this model; auto-generated if not specified.
training_frame		Id of the training data frame.
validation_frame		Id of the validation data frame.
nfolds	0	Number of folds for K-fold cross-validation (0 to disable or >= 2).
keep_cross_validation_predictions	false	Whether to keep the predictions of the cross-validation models.
keep_cross_validation_fold_assignment	false	Whether to keep the cross-validation fold assignment.
score_each_iteration	false	Whether to score during each iteration of model training.
score_tree_interval	0	Score the model after every so many trees. Disabled if set to 0.
fold_assignment	AUTO	Cross-validation fold assignment scheme, if fold_column is not specified. The 'Stratified' option will stratify the folds based on the response variable, for classification problems.
fold_column		Column with cross-validation fold index assignment per observation.
response_column	MARGEN_BRUTO	Response variable column.
ignored_columns		Names of columns to ignore for training.
ignore_const_cols	true	Ignore constant columns.
offset_column		Offset column. This will be added to the combination of columns before applying the link function.
weights_column		Column with observation weights. Giving some observation a weight of zero is equivalent to excluding it from the dataset; giving an observation a relative weight of 2 is equivalent to repeating that row twice. Negative weights are not allowed. Note: Weights are per-row observation weights and do not increase the size of the data frame. This is typically the number of times a row is repeated, but non-integer values are supported as well. During training, rows with higher weights matter more, due to the larger loss function pre-factor.
balance_classes	false	Balance training data class counts via over/under-sampling (for imbalanced data).
class_sampling_factors		Desired over/under-sampling ratios per class (in lexicographic order). If not specified, sampling factors will be automatically computed to obtain class balance during training. Requires balance_classes.
max_after_balance_size	5	Maximum relative size of the training data after balancing class counts (can be less than 1.0). Requires balance_classes.
max_confusion_matrix_size	20	[Deprecated] Maximum size (# classes) for confusion matrices to be printed in the Logs
max_hit_ratio_k	0	Max. number (top K) of predictions to use for hit ratio computation (for multi-class only, 0 to disable)
ntrees	46	Number of trees.
max_depth	16	Maximum tree depth.
min_rows	100	Fewest allowed (weighted) observations in a leaf.
nbins	20	For numerical columns (real/int), build a histogram of (at least) this many bins, then split at the best point
nbins_top_level	1024	For numerical columns (real/int), build a histogram of (at most) this many bins at the root level, then decrease by factor of two per level

Parameter	Value	Description
nbins_cats	1024	For categorical columns (factors), build a histogram of this many bins, then split at the best point. Higher values can lead to more overfitting.
r2_stopping	1.7976931348623157e+308	r2_stopping is no longer supported and will be ignored if set - please use stopping_rounds, stopping_metric and stopping_tolerance instead. Previous version of H2O would stop making trees when the R^2 metric equals or exceeds this
stopping_rounds	0	Early stopping based on convergence of stopping_metric. Stop if simple moving average of length k of the stopping_metric does not improve for k:=stopping_rounds scoring events (0 to disable)
stopping_metric	AUTO	Metric to use for early stopping (AUTO: logloss for classification, deviance for regression)
stopping_tolerance	0.001	Relative tolerance for metric-based stopping criterion (stop if relative improvement is not at least this much)
max_runtime_secs	0	Maximum allowed runtime in seconds for model training. Use 0 to disable.
seed	8764811954202269000	Seed for pseudo random number generator (if applicable)
build_tree_one_node	false	Run on one node only; no network overhead but fewer cpus used. Suitable for small datasets.
learn_rate	0.1	Learning rate (from 0.0 to 1.0)
learn_rate_annealing	1	Scale the learning rate by this factor after each tree (e.g., 0.99 or 0.999)
distribution	laplace	Distribution function
quantile_alpha	0.5	Desired quantile for Quantile regression, must be between 0 and 1.
tweedie_power	1.5	Tweedie power for Tweedie regression, must be between 1 and 2.
huber_alpha	0.9	Desired quantile for Huber/M-regression (threshold between quadratic and linear loss, must be between 0 and 1).
checkpoint		Model checkpoint to resume training with.
sample_rate	0.8	Row sample rate per tree (from 0.0 to 1.0)
sample_rate_per_class		A list of row sample rates per class (relative fraction for each class, from 0.0 to 1.0), for each tree
col_sample_rate	0.7	Column sample rate (from 0.0 to 1.0)
col_sample_rate_change_per_level	1	Relative change of the column sampling rate for every level (from 0.0 to 2.0)
col_sample_rate_per_tree	1	Column sample rate per tree (from 0.0 to 1.0)
min_split_improvement	0.00001	Minimum relative improvement in squared error reduction for a split to happen
histogram_type	AUTO	What type of histogram to use for finding optimal split points
max_abs_leafnode_pred	1.7976931348623157e+308	Maximum absolute value of a leaf node prediction
pred_noise_bandwidth	0	Bandwidth (sigma) of Gaussian multiplicative noise ~N(1,sigma) for tree node predictions
categorical_encoding	AUTO	Encoding scheme for categorical features
calibrate_model	false	Use Platt Scaling to calculate calibrated class probabilities. Calibration can provide more accurate estimates of class probabilities.
calibration_frame		Calibration frame for Platt Scaling
custom_metric_func		Reference to custom evaluation function, format: `language:keyName=funcName`

The caret R package

El paquete de caret (Classification And Regresion Training), es un conjunto de funciones que intentan simplificar el proceso para crear modelos predictivos. Estas funciones actúan como interfaz para decenas de métodos complejos de clasificación y regresión.

Utilizar este paquete en lugar de las funciones originales de los métodos presenta dos ventajas:

- Permite utilizar un código unificado para aplicar reglas de clasificación muy distintas, implementadas en distintos paquetes.
- Es más fácil poner en práctica algunos procedimientos usuales en problemas de clasificación. Por ejemplo, hay funciones específicas para dividir la muestra en datos de entrenamiento y datos de test o para ajustar parámetros mediante validación cruzada.

Python

Se trata de un lenguaje de programación interpretado y multiparadigma, ya que soporta orientación a objetos, programación interactiva y, en menor medida, programación funcional. Uso tipado y dinámico y es multiplataforma.

Es un lenguaje de programación con numerosas librerías creadas para el análisis de datos y con una integración con aplicaciones con Mongo DB, Hadoop o Pentaho. Ademas, tiene una curva de aprendizaje fácil y rápidalo que le convierte en un lenguaje de gran calidad para el análisis de datos.

R

Entorno y lenguaje de programación multiparadigma, orientado a objetos, vectorial y multiplataforma, con enfoque al análisis estadístico. Por ello es muy preciso y exacto para el análisis de datos.

Proporciona un amplio abanico de herramientas estadísticas (modelos lineales y no lineales, test estadísticos, análisis de series temporales, algoritmos de clasificación y agrupamiento, etc.) y gráficas.

Como contra en R, está su curva de aprendizaje, que suele ser más lenta y complicada si la comparamos con la de Python.

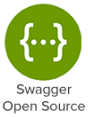
Studio-3T

IDE para la gestión de MongoDB.

Postman

Está compuesto por diferentes herramientas y utilidades gratuitas (en la versión free) que permiten realizar diferentes tareas dentro del mundo API Rest: creación de APIs, elaboración de test para validar el comportamiento de las APIs, posibilidad de crear entornos de trabajo diferentes (con variables locales y globales).
En el caso de nuestro proyecto se utilizó como una herramienta para hacer peticiones APIs y generar colecciones de peticiones que permitan validar su comportamiento de manera rápida y sencilla.

Swagger



Se trata de un framework (especificación e implementación) completo, para diseñar, construir, documentar y consumir servicios web RESTful. El conjunto de herramientas de Swagger incluye soporte para documentación automatizada, generación de código y generación de casos de prueba. El objetivo de Swagger es que la documentación de API RESTFul se vaya actualizando cada vez que se realicen cambios en el servidor.

Android Studio

IDE oficial para aplicaciones Android. Proporciona las herramientas más rápidas para crear apps en todas las clases de dispositivos Android.

MongoDB

Es un sistema de Base de datos NoSQL orientada a documentos, desarrollado bajo el concepto de código abierto. Esto quiere decir que en lugar de guardar los datos en registros los guarda en documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON.

Power BI

Herramienta de Business Inteligence (BI) que incorpora la suite de productividad Office 365. Esta herramienta permite analizar e interactuar con

una gran cantidad de datos dentro de Excel. Permite obtener de manera fácil, información del valor de los datos, trabajando desde Excel para analizarlos y visualizarlos de forma autónoma.

Filezilla

Programa gratuito, para dotar a nuestro sistema Windows de capacidades para la distribución de archivos por medio de FTP (File Transfer Protocol).

Putty

Emulador gratuito de terminal que soporta SSH y muchos otros protocolos. Utilizado para conectar a un servidor Unix o Linux a través de SSH.Ofrece una interfaz gráfica de configuración muy sencilla que integra múltiples opciones.

Script R para desarrollo de los modelos

A Continuación reflejamos la dinámica de trabajo seguida para cada uno de los modelos, el ejemplo lo reflejamos con el modelo de NEGATIVOS.

Código R

```
#Algoritmo de Negativos Clientes NEGATIVE

h2o.init(port=54844,max_mem_size = "128g")

h2o_neg= h2o.uploadFile("DATA_VAR_NEGATIVOS.csv")
str(h2o_neg)

h2o_neg[,"COD_SEGMENTO_1"] <- as.factor(h2o_neg[,"COD_SEGMENTO_1"])
h2o_neg[,"COD_PROVINCIA"] <- as.factor(h2o_neg[,"COD_PROVINCIA"])
h2o_neg[,"SOCGRUPOMOSAIC"] <- as.factor(h2o_neg[,"SOCGRUPOMOSAIC"])
h2o_neg[,"NOTA2V"] <- as.factor(h2o_neg[,"NOTA2V"])
h2o_neg[,"NOTA1"] <- as.factor(h2o_neg[,"NOTA1"])
h2o_neg[,"SOCSEGMENTOSEGUROS"] <- as.factor(h2o_neg[,"SOCSEGMENTOSEGUROS"])

str(h2o_neg)

summary(h2o_neg$MARGEN_BRUTO)


r2 <- h2o.runif(h2o_neg)
train_neg<- h2o_neg[r2 < 0.6,]
test_neg<- h2o_neg[(r2 >= 0.6) & (r2 < 0.9),]
hold_neg<-h2o_neg[r2 >= 0.9,]
response_neg<- "MARGEN_BRUTO"
predictors_neg<- setdiff(names(h2o_neg), response_neg)


#The current version of AutoML trains and cross-validates a default Random Forest, an
Extremely-Randomized Forest, a random grid of Gradient Boosting Machines (GBMs), a random
grid of Deep Neural Nets, a fixed grid of GLMs, and then trains two Stacked Ensemble
models.

automl_h2o_models_neg<- h2o.automl(
  x = predictors_neg,
  y = response_neg,
  training_frame    = train_neg,
  leaderboard_frame = hold_neg,
  max_runtime_secs = 3600,
  max_models=100,
  stopping_metric="MSE",
  seed = 789101234)

model_ids_neg <- as.data.frame(automl_h2o_models_neg@leaderboard$model_id)[,1]
model_ids_neg

[1] "StackedEnsemble_AllModels_0_AutoML_20180407_123326"
[2] "StackedEnsemble_BestOfFamily_0_AutoML_20180407_123326"
[3] "DeepLearning_grid_0_AutoML_20180407_123326_model_2"
[4] "GBM_grid_0_AutoML_20180407_123326_model_8"
[5] "DeepLearning_0_AutoML_20180407_123326"
[6] "DeepLearning_grid_0_AutoML_20180407_123326_model_4"
[7] "GBM_grid_0_AutoML_20180407_123326_model_45"
[8] "GBM_grid_0_AutoML_20180407_123326_model_25"
[9] "DeepLearning_grid_0_AutoML_20180407_123326_model_11"
[10] "GBM_grid_0_AutoML_20180407_123326_model_42"
[11] "GBM_grid_0_AutoML_20180407_123326_model_14"
[12] "GBM_grid_0_AutoML_20180407_123326_model_38"
[13] "GBM_grid_0_AutoML_20180407_123326_model_22"
[14] "DeepLearning_grid_0_AutoML_20180407_123326_model_1"
[15] "GBM_grid_0_AutoML_20180407_123326_model_1"
[16] "GBM_grid_0_AutoML_20180407_123326_model_9"
[17] "GBM_grid_0_AutoML_20180407_123326_model_24"
```

```
[18] "GBM_grid_0_AutoML_20180407_123326_model_20"
[19] "GBM_grid_0_AutoML_20180407_123326_model_43"
[20] "GBM_grid_0_AutoML_20180407_123326_model_27"
[21] "GBM_grid_0_AutoML_20180407_123326_model_19"
[22] "DeepLearning_grid_0_AutoML_20180407_123326_model_12"
[23] "GBM_grid_0_AutoML_20180407_123326_model_0"
[24] "DeepLearning_grid_0_AutoML_20180407_123326_model_7"
[25] "DeepLearning_grid_0_AutoML_20180407_123326_model_16"
[26] "GBM_grid_0_AutoML_20180407_123326_model_4"
[27] "GBM_grid_0_AutoML_20180407_123326_model_6"
[28] "DeepLearning_grid_0_AutoML_20180407_123326_model_10"
[29] "GBM_grid_0_AutoML_20180407_123326_model_44"
[30] "GBM_grid_0_AutoML_20180407_123326_model_28"
[31] "GBM_grid_0_AutoML_20180407_123326_model_35"
[32] "GBM_grid_0_AutoML_20180407_123326_model_30"
[33] "GBM_grid_0_AutoML_20180407_123326_model_2"
[34] "DeepLearning_grid_0_AutoML_20180407_123326_model_6"
[35] "DeepLearning_grid_0_AutoML_20180407_123326_model_15"
[36] "DeepLearning_grid_0_AutoML_20180407_123326_model_3"
[37] "DeepLearning_grid_0_AutoML_20180407_123326_model_13"
[38] "GBM_grid_0_AutoML_20180407_123326_model_40"
[39] "GBM_grid_0_AutoML_20180407_123326_model_34"
[40] "GBM_grid_0_AutoML_20180407_123326_model_37"
[41] "GBM_grid_0_AutoML_20180407_123326_model_3"
[42] "GBM_grid_0_AutoML_20180407_123326_model_26"
[43] "DeepLearning_grid_0_AutoML_20180407_123326_model_0"
[44] "GBM_grid_0_AutoML_20180407_123326_model_36"
[45] "GBM_grid_0_AutoML_20180407_123326_model_32"
[46] "GBM_grid_0_AutoML_20180407_123326_model_15"
[47] "GBM_grid_0_AutoML_20180407_123326_model_21"
[48] "GBM_grid_0_AutoML_20180407_123326_model_41"
[49] "GBM_grid_0_AutoML_20180407_123326_model_17"
[50] "GBM_grid_0_AutoML_20180407_123326_model_16"
[51] "GBM_grid_0_AutoML_20180407_123326_model_18"
[52] "GLM_grid_0_AutoML_20180407_123326_model_0"
[53] "DeepLearning_grid_0_AutoML_20180407_123326_model_8"
[54] "DeepLearning_grid_0_AutoML_20180407_123326_model_9"
[55] "GBM_grid_0_AutoML_20180407_123326_model_5"
[56] "DeepLearning_grid_0_AutoML_20180407_123326_model_5"
[57] "DeepLearning_grid_0_AutoML_20180407_123326_model_14"
[58] "GBM_grid_0_AutoML_20180407_123326_model_33"
[59] "XRT_0_AutoML_20180407_123326"
[60] "GBM_grid_0_AutoML_20180407_123326_model_29"
[61] "DRF_0_AutoML_20180407_123326"
[62] "GBM_grid_0_AutoML_20180407_123326_model_12"
[63] "GBM_grid_0_AutoML_20180407_123326_model_11"
[64] "GBM_grid_0_AutoML_20180407_123326_model_7"
[65] "GBM_grid_0_AutoML_20180407_123326_model_23"
[66] "GBM_grid_0_AutoML_20180407_123326_model_39"
[67] "GBM_grid_0_AutoML_20180407_123326_model_31"
[68] "GBM_grid_0_AutoML_20180407_123326_model_13"
[69] "GBM_grid_0_AutoML_20180407_123326_model_10"
```

Al final de este proceso, siempre seleccionamos el primer modelo que pueda ser explicativo, ya que esto imprescindible para que pueda ser aceptado por el departamento Técnico Actuarial. En el ejemplo anterior el mejor modelo fue el 4.

```
[4] "GBM_grid_0_AutoML_20180407_123326_model_8"
```

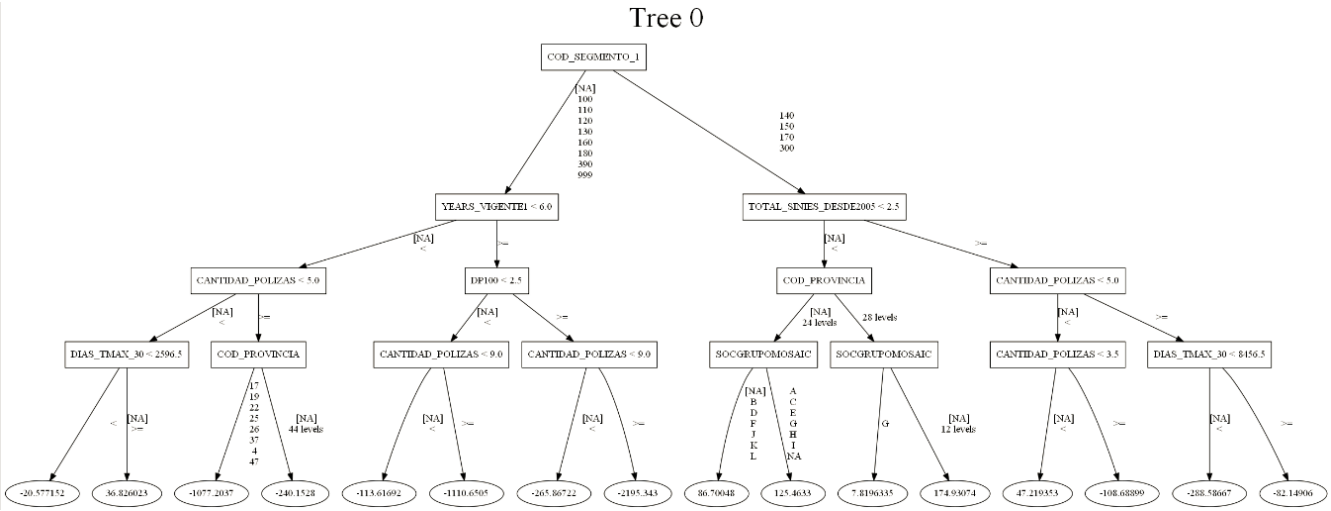
Script Para aplicar las predicciones y evaluar el modelo

```
se_ind_nc<- h2o.getModel(grep( "GBM_grid_0_AutoML_20180403_121621_model_0" , model_ids_nc,
value = TRUE)[1])
predictions_posneg<-as.data.frame(h2o.predict(object = se_ind_nc
                                              ,newdata = test_nc))
predictions posneg1<-as.data.frame(h2o.predict(object = se ind nc
                                               ,newdata = h2o_notr))
```

```
test test<-as.data.frame(test_nc)
test1<-df_var_noclientes
test_test$predictions<-predictions_posneg$predict
test1$predictions<-predictions_posneg1$predict
test_test$diferencias<-(test_test$predictions-
test_test$MARGEN_BRUTO)/test_test$MARGEN_BRUTO
test test$porcentaje <- with(test test, ifelse(diferencias<1, "<1%",ifelse((diferencias>1&
diferencias<5), "1-5%",ifelse((diferencias>5& diferencias<10), "5-10%",

+ ifelse((diferencias>10& diferencias<15), "10-15%",">15%")))))
table(test_test$porcentaje)
mean(test_test$predictions)
mean(test test$MARGEN BRUTO)
max(test_test$predictions)
max(test_test$MARGEN_BRUTO)
min(test_test$predictions)
min(test_test$MARGEN_BRUTO)
mean(test1$predictions)
mean(test1$MARGEN_BRUTO)
max(test1$predictions)
max(test1$MARGEN_BRUTO)
min(test1$predictions)
min(test1$MARGEN_BRUTO)
write.csv2(test_test, "NO_CLIENTES03042018.csv",row.names=FALSE)
h2o_all= as.h2o(DATA_MAYORIGUAL18)
predictions_posneg_h2o<-as.data.frame(h2o.predict(object = se_ind_imp
                                                  ,newdata = h2o_all))
View(predictions_posneg_h2o)
test_all<-DATA_MAYORIGUAL18
test_all$predictions<-predictions_posneg_h2o$predict
test_all$diferencias<-(test_all$predictions- test_all$MARGEN.BRUTO)/test_all$MARGEN.BRUTO
test_all$porcentaje <- with(test_all, ifelse(diferencias<1, "<1%",ifelse((diferencias>1&
diferencias<5), "1-5%",ifelse((diferencias>5& diferencias<10), "5-10%",
```

A partir de este modelo, se aplican las predicciones, se exporta en formato POJO y MOJO, para pintar los árboles y que estos también puedan ser evaluados por los actuarios y cuando ya se selecciona como definitivo se pasa a formar parte de la API que se publica en Azure.

Ejemplo de Arboles generados, utilizando H2o y la apliación “Graphiviz”.



Control de Ejecuciones:

Jobs

Type	Destination	Description	Start Time	End Time	Run Time	Status
Frame	DATACLIENTES_sid_884c_2	Parse	2018-01-11 10:29:04	2018-01-11 10:29:10	00:00:05.975	DONE
Model	GBM_model_Oracle_1515662939536_1	GBM	2018-01-11 10:30:54	2018-01-11 10:31:11	00:00:16.843	DONE
Model	GBM_model_Oracle_1515662939536_2	GBM	2018-01-11 11:44:15	2018-01-11 11:46:54	00:02:39.583	DONE
Model	GBM_model_Oracle_1515662939536_3	GBM	2018-01-11 12:01:53	2018-01-11 12:03:32	00:01:39.36	DONE
Grid	depth_grid	GBM Grid Search	2018-01-11 14:09:11	2018-01-11 14:09:58	00:00:46.780	CANCELLED
Grid	depth_grid	GBM Grid Search	2018-01-11 14:13:06	2018-01-11 15:19:46	01:06:40.456	DONE

La ejecución de 1Hora se corresponde al Grid del modelo 2 con los siguientes parámetros:

```
grid <- h2o.grid(
 ## hyper parameters
 hyper_params = hyper_params,

 ## full Cartesian hyper-parameter search
 search_criteria = list(strategy = "Cartesian"),

 ## which algorithm to run
 algorithm="gbm",

 ## identifier for the grid, to later retrieve it
 grid_id="depth_grid",

 ## standard model parameters
 x = predictors,
 y = response,
 training_frame = train,
 validation_frame = valid,

 ## more trees is better if the learning rate is small enough
 ## here, use "more than enough" trees - we have early stopping
 ntrees = 10000,

 ## smaller learning rate is better
 ## since we have learning_rate_annealing, we can afford to start with a bigger learning rate
 learn_rate = 0.05,

 ## learning rate annealing: learning_rate shrinks by 1% after every tree
 ## (use 1.00 to disable, but then lower the learning_rate)
 learn_rate_annealing = 0.99,
```

```
## sample 80% of rows per tree
sample_rate = 0.8,

## sample 80% of columns per split
col_sample_rate = 0.8,

## fix a random number generator seed for reproducibility
seed = 77777,

## early stopping once the validation AUC doesn't improve by at least 0.01% for 5 consecutive scoring events
stopping_rounds = 5,
stopping_tolerance = 1e-4,
stopping_metric = "MSE",

## score every 10 trees to make early stopping reproducible (it depends on the scoring interval)
score_tree_interval = 10
)
```

Jobs

Type	Destination	Description	Start Time	End Time	Run Time	Status
Frame	DATACLIENTES_sid_884c_2	Parse	2018-01-11 10:29:04	2018-01-11 10:29:10	00:00:05.975	DONE
Model	GBM_model_Oracle_1515662939536_1	GBM	2018-01-11 10:30:54	2018-01-11 10:31:11	00:00:16.843	DONE
Model	GBM_model_Oracle_1515662939536_2	GBM	2018-01-11 11:44:15	2018-01-11 11:46:54	00:02:39.583	DONE
Model	GBM_model_Oracle_1515662939536_3	GBM	2018-01-11 12:01:53	2018-01-11 12:03:32	00:01:39.36	DONE
Grid	depth_grid	GBM Grid Search	2018-01-11 14:09:11	2018-01-11 14:09:58	00:00:46.780	CANCELLED
Grid	depth_grid	GBM Grid Search	2018-01-11 14:13:06	2018-01-11 15:19:46	01:06:40.456	DONE
Grid	final_grid	GBM Grid Search	2018-01-11 17:10:52	2018-01-11 17:53:02	00:42:09.570	DONE
Model	GBM_model_Oracle_1515662939536_29	GBM	2018-01-11 18:00:31	2018-01-11 18:11:15	00:10:44.474	DONE

Ejecución1 por el puerto 54777

Jobs

Type	Destination	Description	Start Time	End Time	Run Time	Status
Frame	DATACLIENTES_sid_91cd_2	Parse	2018-01-11 21:59:34	2018-01-11 21:59:39	00:00:04.792	DONE
Model	GBM_model_Oracle_1515704237993_1	GBM	2018-01-11 22:01:10	2018-01-11 22:01:32	00:00:21.455	DONE
Model	GBM_model_Oracle_1515704237993_2	GBM	2018-01-11 22:11:52	2018-01-11 22:14:41	00:02:48.998	DONE
Model	GBM_model_Oracle_1515704237993_3	GBM	2018-01-11 22:31:55	2018-01-11 22:32:55	00:00:59.219	RUNNING

Refresh

CS

```
getModels
```

83ms

Error evaluating cell

Error calling GET /3/Models

Could not connect to H2O. Your H2O cloud is currently unresponsive.

TOGGLE STACK TRACE

Ejecución 2

```
> h2o.init(port=54777,max_mem_size = "96g")
 Connection successful!

R is connected to the H2O cluster:
    H2O cluster uptime:         3 minutes 39 seconds
    H2O cluster version:        3.16.0.2
    H2O cluster version age:    1 month and 12 days
    H2O cluster name:           H2O_started_from_R_nilley_bag098
    H2O cluster total nodes:    1
    H2O cluster total memory:   84.12 GB
    H2O cluster total cores:    32
    H2O cluster allowed cores:  32
    H2O cluster healthy:        TRUE
    H2O Connection ip:          localhost
    H2O Connection port:        54777
    H2O Connection proxy:       NA
    H2O Internal Security:      FALSE
    H2O API Extensions:         Algos, AutoML, Core V3, Core V4
    R Version:                  Oracle Distribution of R version 3.2.0  (--)

> h2o_df = h2o.uploadFile("DATACLIENTES.csv")
  |=========================================================================================| 100%
```

Jobs

Type	Destination	Description	Start Time	End Time	Run Time	Status
Frame	DATACLIENTES_sid_9c4e_2	Parse	2018-01-12 05:16:26	2018-01-12 05:16:31	00:00:05.83	DONE

Ejecución Modelo 1 – Simple con parámetros por defecto

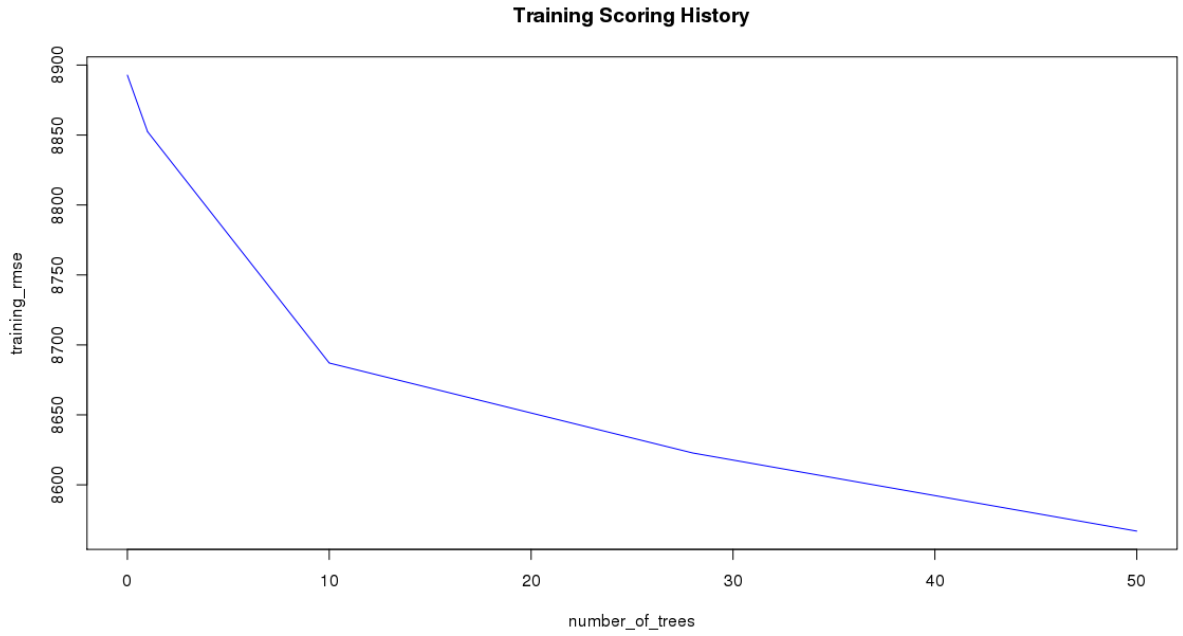
```
gbm_basicmodel<- h2o.gbm(x = predictors, y = response, training_frame = train)
 |=========================================================================================| 100%
```

Model	GBM_model_Oracle_1515730253930_1	GBM	2018-01-12 05:19:44	2018-01-12 05:20:01	00:00:17.507	DONE

```
Model Details:
==============

H2ORegressionModel: gbm
Model ID:  GBM_model_Oracle_1515730253930_1
Model Summary:
  number_of_trees number_of_internal_trees model_size_in_bytes min_depth m
ax_depth
1              50                        50               22114         5
5
  mean_depth min_leaves max_leaves mean_leaves
1    5.00000         10         32    27.20000


H2ORegressionMetrics: gbm
```

```
** Reported on training data. **

MSE:  73392145
RMSE:  8566.922
MAE:  1374.505
RMSLE:  NaN
Mean Residual Deviance :  73392145
```

plot(gbm_basicmodel)



plot(gbm_basicmodel, timestep = "number_of_trees", metric = "mae")

gbm_basicmodel_performance

#51305037

Model

Model ID: GBM_model_Oracle_1515730253930_1

Algorithm: Gradient Boosting Machine

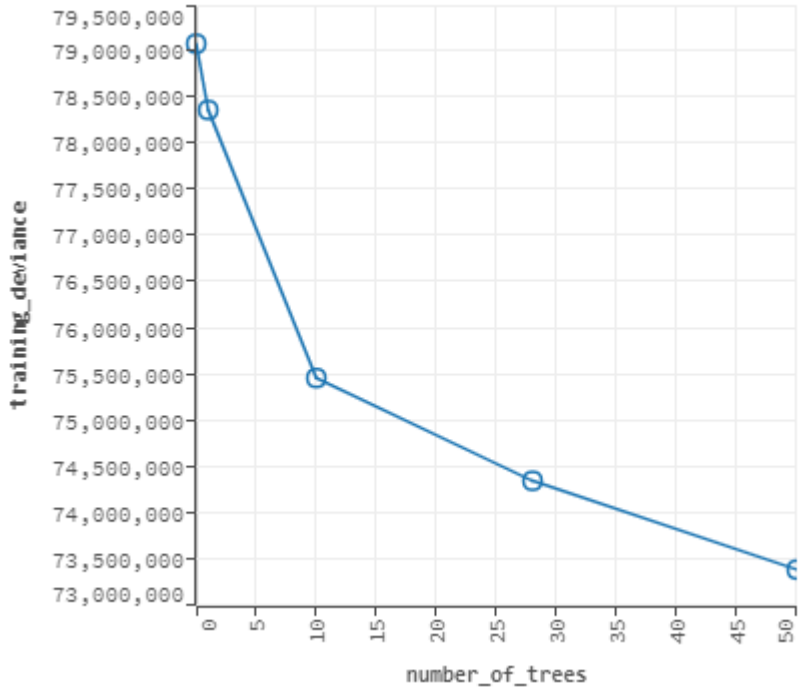
Actions: Refresh | Predict... | Download POJO | Download Model Deployment Package (MOJO) | Export | Inspect | Delete | Download Gen Model

MODEL PARAMETERS

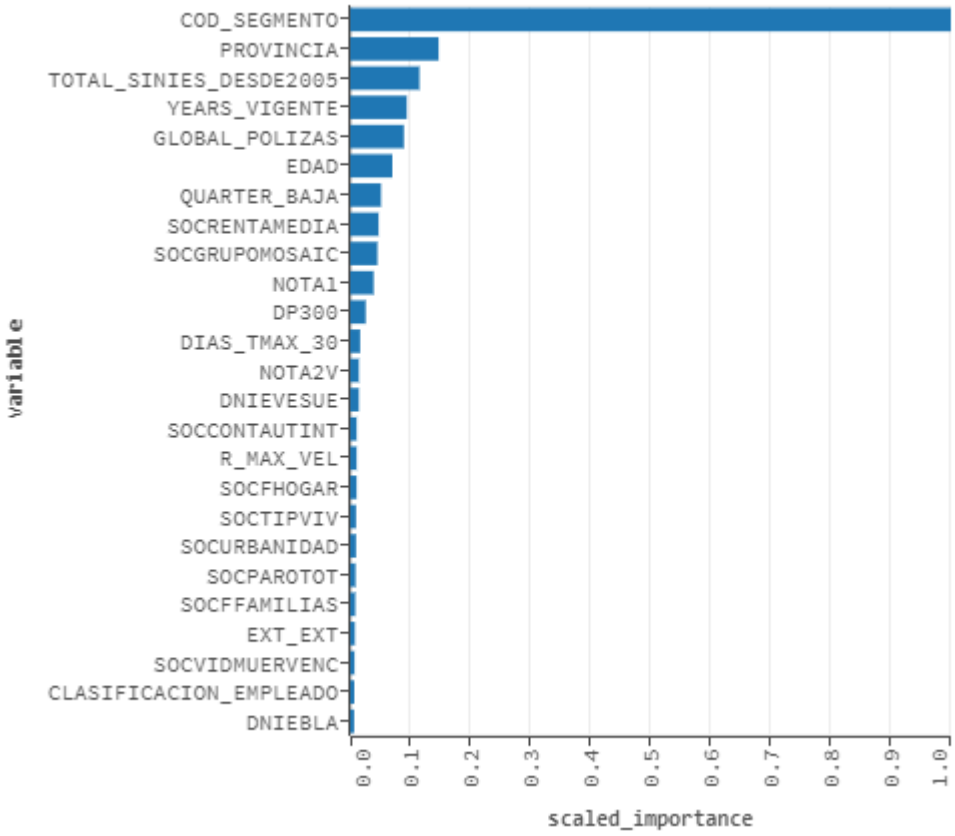
Show all parameters

Parameter	*Value*	*Description*
model_id	GBM_model_Oracle_1515730253930_1	Destination id for this model; auto-generated if not specified.
training_frame	train.hex	Id of the training data frame.
response_column	MARGEN_BRUTO	Response variable column.
ignored_columns		Names of columns to ignore for training.
seed	-330442701210250430	Seed for pseudo random number generator (if applicable)
distribution	gaussian	Distribution function

SCORING HISTORY - DEVIANCE



▾ VARIABLE IMPORTANCES



▾ OUTPUT

cross_validation_models	·
cross_validation_predictions	·
cross_validation_holdout_predictions_frame_id	·
cross_validation_fold_assignment_frame_id	·
model_category	Regression
validation_metrics	·
cross_validation_metrics	·
cross_validation_metrics_summary	·
status	·
start_time	1515730784154
end_time	1515730801618
run_time	17464
init_f	926.992089

▾ OUTPUT - MODEL SUMMARY

number_of_trees	50
number_of_internal_trees	50
model_size_in_bytes	22114
min_depth	5
max_depth	5
mean_depth	5.0
min_leaves	10
max_leaves	32
mean_leaves	27.2000

▾ OUTPUT - SCORING HISTORY

timestamp	duration	number_of_trees	training_rmse	training_mae	training_deviance
2018-01-12 04:19:44	0.040 sec	0	8892.7951	1775.7939	79081803.9061
2018-01-12 04:19:49	5.029 sec	1	8852.3789	1682.4185	78364611.8314
2018-01-12 04:19:53	9.198 sec	10	8687.0617	1377.1881	75465040.1694
2018-01-12 04:19:57	13.377 sec	28	8622.7797	1364.7049	74352329.5667
2018-01-12 04:20:01	17.479 sec	50	8566.9215	1374.5054	73392144.8012

▾ OUTPUT - TRAINING_METRICS

model	GBM_model_Oracle_1515730253930_1
model_checksum	6818052531613759488
frame	train.hex
frame_checksum	-793620439018086784
description	·
model_category	Regression
scoring_time	1515730801614
predictions	·
MSE	73392144.801199
RMSE	8566.921548
nobs	1135059
custom_metric_name	·
custom_metric_value	0
r2	0.071947
mean_residual_deviance	73392144.801199
mae	1374.505404
rmsle	NaN

▾ OUTPUT - VARIABLE IMPORTANCES

variable	relative_importance	scaled_importance	percentage
COD_SEGMENTO	17768975958016.0	1.0	0.5228
PROVINCIA	2612428013568.0	0.1470	0.0769
TOTAL_SINIES_DESDE2005	2046358061056.0	0.1152	0.0602
YEARS_VIGENTE	1671594901504.0	0.0941	0.0492
GLOBAL_POLIZAS	1584816848896.0	0.0892	0.0466
EDAD	1243026554880.0	0.0700	0.0366
QUARTER_BAJA	908612075520.0	0.0511	0.0267
SOCRENTAMEDIA	835919872000.0	0.0470	0.0246
SOCGRUPOMOSAIC	802616705024.0	0.0452	0.0236
NOTA1	690572427264.0	0.0389	0.0203
DP300	456215986176.0	0.0257	0.0134
DIAS_TMAX_30	289520582656.0	0.0163	0.0085
NOTA2V	252886548480.0	0.0142	0.0074
DNIEVESUE	252172042240.0	0.0142	0.0074
SOCCONTAUTINT	186849230848.0	0.0105	0.0055
R_MAX_VEL	186449149952.0	0.0105	0.0055
SOCFHOGAR	185220661248.0	0.0104	0.0054
SOCTIPVIV	169272180736.0	0.0095	0.0050
SOCURBANIDAD	168504705024.0	0.0095	0.0050
SOCPAROTOT	155497922560.0	0.0088	0.0046
SOCFFAMILIAS	148775911424.0	0.0084	0.0044
EXT_EXT	135333986304.0	0.0076	0.0040
SOCVIDMUERVENC	124624764928.0	0.0070	0.0037

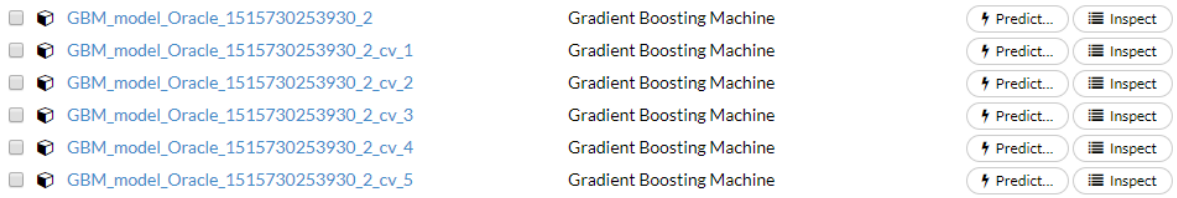
CLASIFICACION_EMPLEADO	118076276736.0	0.0066	0.0035
DNIEBLA	112940498944.0	0.0064	0.0033
SOCCONTBANCO	103352410112.0	0.0058	0.0030
SOCTRANSITORIEDAD	103070990336.0	0.0058	0.0030
SOCCONTAUTBANCO	96340566016.0	0.0054	0.0028
DTORMENTA	82060427264.0	0.0046	0.0024
COMU_AUTONOMA	72978776064.0	0.0041	0.0021
SOCCOCHES1	65686192128.0	0.0037	0.0019
SOCMOTRAPI	54776201216.0	0.0031	0.0016
QUARTER_ALTA	51439792128.0	0.0029	0.0015
SOCSEGMEDEMP	45061152768.0	0.0025	0.0013
SOCMOTPRECIO	35705339904.0	0.0020	0.0011
SOCMOTNECE	34901360640.0	0.0020	0.0010
SOCCONTM5	19615858688.0	0.0011	0.0006
SOCTEROPC	17381537792.0	0.0010	0.0005
CONT_VIGOR	14071964672.0	0.0008	0.0004
SOCSEGMENTOSEGUROS	11707280384.0	0.0007	0.0003
DROCIO	11558287360.0	0.0007	0.0003
SOCANTMA5	10514080768.0	0.0006	0.0003
SOCCALIDADVIV	10272920576.0	0.0006	0.0003
SOCMOTOFER	10093560832.0	0.0006	0.0003
SOCCONTMEDIADOR	6705200640.0	0.0004	0.0002
SOCESTRES	5553901568.0	0.0003	0.0002
SOCMOTCONF	5027309568.0	0.0003	0.0001
SOCTRCF	4848854016.0	0.0003	0.0001
SOCMOTSERV	3923047680.0	0.0002	0.0001
SOCCONTAUTMEDIADOR	2873139200.0	0.0002	0.0001
SOCCONTCIA	2788801536.0	0.0002	0.0001
DENSIDAD	411612992.0	0.0	0.0
SOCSEGMEDPARC	0	0	0
SOCCOMPRAINTERNET	0	0	0
HABITANTES	0	0	0

Modelo 2

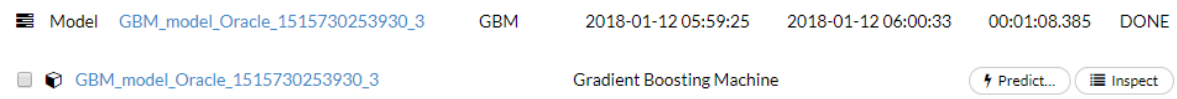

```
gbm_basic2@model$cross_validation_metrics_summary
Cross-Validation Metrics Summary:
                             mean           sd  cv_1_valid  cv_2_valid  cv_3_valid
mae                     1363.6798    2.8719885   1356.0482   1367.6731   1363.9877
mean_residual_deviance 6.9587376E7    8029373.5 5.0892472E7 6.4109648E7 7.6858848E7
mse                    6.9587376E7    8029373.5 5.0892472E7 6.4109648E7 7.6858848E7
r2                      0.05187674 0.0061448948  0.06601983 0.056338225 0.046964202
residual_deviance      6.9587376E7    8029373.5 5.0892472E7 6.4109648E7 7.6858848E7
rmse                       8312.571    494.23227    7133.896     8006.85    8766.918
rmsle                           0.0          NaN         NaN         NaN         NaN
                         cv_4_valid  cv_5_valid
mae                       1366.4972   1364.1932
mean_residual_deviance 8.3933056E7 7.2142864E7
mse                    8.3933056E7 7.2142864E7
r2                     0.040569972 0.049491465
residual_deviance      8.3933056E7 7.2142864E7
rmse                       9161.499    8493.695
rmsle                           NaN         NaN
```

h2o.mse(h2o.performance(gbm_basic2, xval = TRUE))

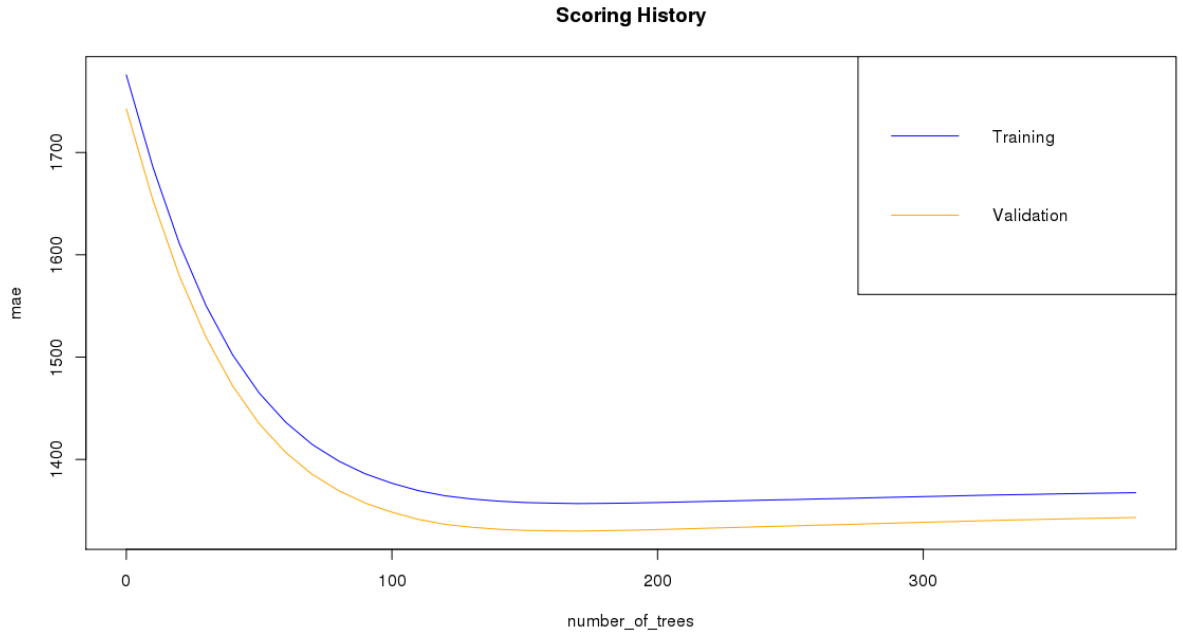
#69592305

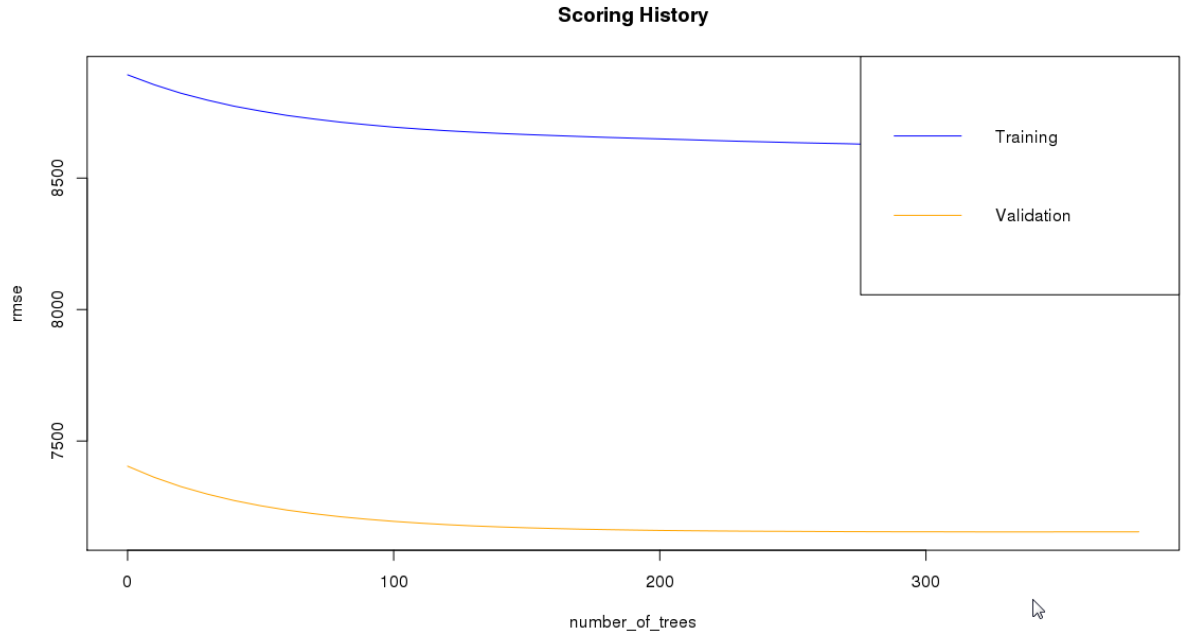


Modelo 3




```
> h2o.mse(h2o.performance(gbm_tunned_me, valid = TRUE))
[1] 51186831
```

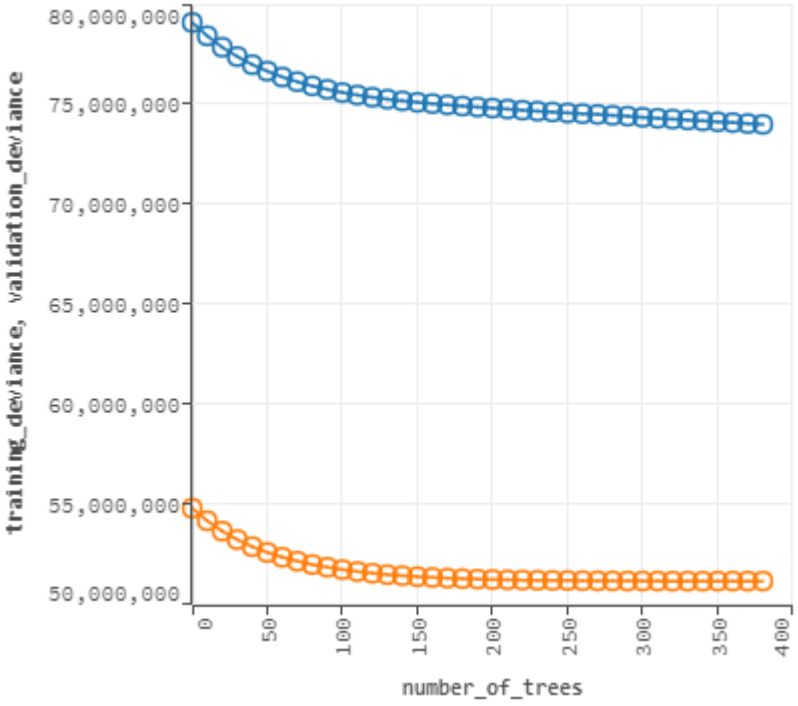

Scoring History
mae
1400
1500
1600
1700
0
100
200
300
number_of_trees
Training
Validation


Scoring History
rmse
7500
8000
8500
0
100
200
300
number_of_trees
Training
Validation

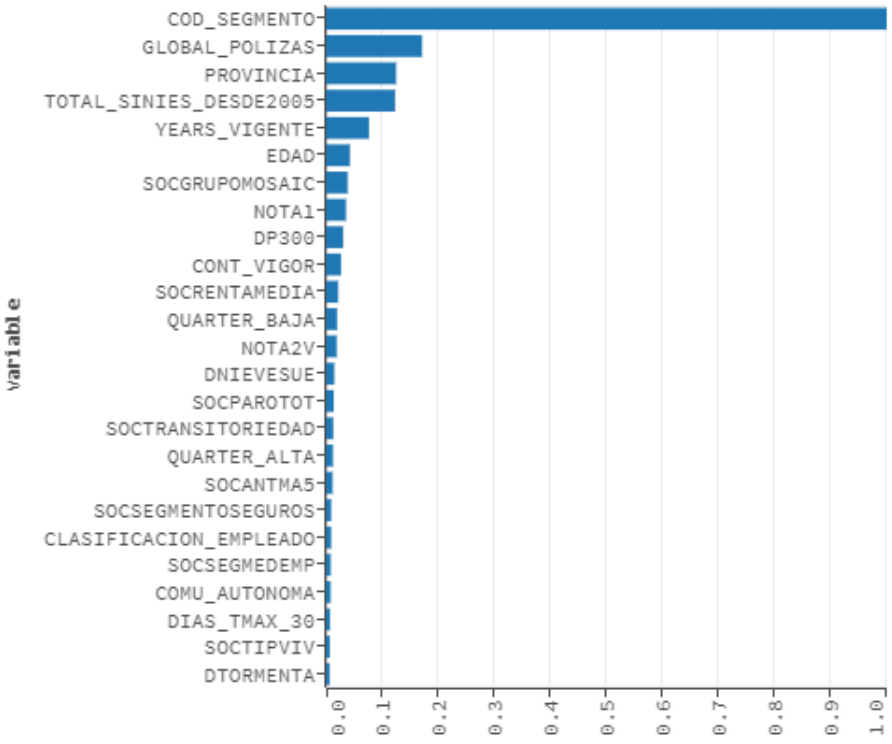
Parametros

Parameter	Value	Description
model_id	GBM_model_Oracle_1515730253930_3	Destination id for this model; auto-generated if not specified.
training_frame	train.hex	Id of the training data frame.
validation_frame	valid.hex	Id of the validation data frame.
score_tree_interval	10	Score the model after every so many trees. Disabled if set to 0.
response_column	MARGEN_BRUTO	Response variable column.
ignored_columns		Names of columns to ignore for training.
ntrees	10000	Number of trees.
stopping_rounds	5	Early stopping based on convergence of stopping_metric. Stop if simple moving average of length k of the stopping_metric does not improve for k:=stopping_rounds scoring events (0 to disable)
stopping_metric	MSE	Metric to use for early stopping (AUTO: logloss for classification, deviance for regression)
stopping_tolerance	0.0001	Relative tolerance for metric-based stopping criterion (stop if relative improvement is not at least this much)
seed	8888	Seed for pseudo random number generator (if applicable)
learn_rate	0.01	Learning rate (from 0.0 to 1.0)
distribution	gaussian	Distribution function
sample_rate	0.8	Row sample rate per tree (from 0.0 to 1.0)
col_sample_rate	0.8	Column sample rate (from 0.0 to 1.0)

▾ SCORING HISTORY - DEVIANCE



▾ VARIABLE IMPORTANCES



▾ OUTPUT

cross_validation_models	·
cross_validation_predictions	·
cross_validation_holdout_predictions_frame_id	·
cross_validation_fold_assignment_frame_id	·
model_category	Regression
cross_validation_metrics	·
cross_validation_metrics_summary	·
status	·
start_time	1515733165202
end_time	1515733233584
run_time	68382
init_f	926.992089

▾ OUTPUT - MODEL SUMMARY

number_of_trees	380
number_of_internal_trees	380
model_size_in_bytes	170423
min_depth	5
max_depth	5
mean_depth	5.0
min_leaves	16
max_leaves	32
mean_leaves	27.6053

▾ OUTPUT - SCORING HISTORY

timestamp	duration	number_of_trees	training_rmse	training_mae	training_deviance	validation_rmse	validation_mae	validation_deviance
2018-01-12 04:59:25	0.002 sec	0	8892.7951	1775.7939	79081803.9061	7404.2503	1742.7509	54822922.7306
2018-01-12 04:59:29	3.874 sec	10	8855.0654	1685.7846	78412182.5493	7361.9051	1653.4870	54197646.7040
2018-01-12 04:59:31	5.966 sec	20	8822.9260	1610.6822	77844022.8424	7326.6844	1579.1500	53680303.5952
2018-01-12 04:59:32	7.702 sec	30	8796.5140	1550.3231	77378658.5942	7297.9653	1519.4248	53260298.1269
2018-01-12 04:59:34	9.434 sec	40	8773.5323	1502.3141	76974868.1846	7273.6170	1471.9497	52905503.8366
2018-01-12 04:59:36	11.182 sec	50	8754.9532	1464.9633	76649205.5084	7253.7779	1435.0184	52617293.9758
2018-01-12 04:59:38	12.904 sec	60	8738.1719	1436.3584	76355648.1997	7237.1357	1406.7835	52376132.4919
2018-01-12 04:59:39	14.641 sec	70	8724.8660	1414.5713	76123286.2138	7223.5696	1385.3554	52179957.1299

2018-01-12 04:59:41	16.381 sec	80	8712.7622	1398.3290	75912224.7340	7211.9115	1369.4241	52011667.9251
2018-01-12 04:59:43	18.148 sec	90	8702.9001	1385.8621	75740469.6056	7202.5037	1357.2841	51876059.5889
2018-01-12 04:59:45	19.883 sec	100	8693.7081	1376.7260	75580560.0056	7194.3612	1348.3997	51758833.0765
2018-01-12 04:59:46	21.634 sec	110	8686.4837	1369.4394	75454999.8139	7187.5963	1341.3287	51661540.3787
2018-01-12 04:59:48	23.402 sec	120	8680.1455	1364.4444	75344926.3408	7181.8313	1336.5486	51578700.2074
2018-01-12 04:59:50	25.132 sec	130	8674.6972	1361.3509	75250371.5619	7176.9996	1333.6983	51509323.0336
2018-01-12 04:59:52	26.880 sec	140	8669.8045	1359.2479	75165509.5054	7173.1830	1331.8028	51454554.9813
2018-01-12 04:59:53	28.582 sec	150	8665.6371	1357.8597	75093266.8456	7169.8557	1330.6138	51406830.9570
2018-01-12 04:59:55	30.289 sec	160	8661.8938	1357.2839	75028404.4647	7167.1254	1330.2277	51367687.0052
2018-01-12 04:59:57	32.059 sec	170	8658.2900	1356.8381	74965986.1713	7164.7562	1329.9604	51333732.1065
2018-01-12 04:59:59	33.832 sec	180	8654.8544	1357.0630	74906503.9935	7162.9384	1330.3480	51307685.8340
2018-01-12 05:00:00	35.595 sec	190	8651.8683	1357.3568	74854824.8314	7161.3579	1330.7957	51285046.6880
2018-01-12 05:00:02	37.309 sec	200	8649.0752	1357.8627	74806501.4949	7159.9144	1331.4207	51264374.8469
2018-01-12 05:00:04	39.004 sec	210	8646.2009	1358.4673	74756790.2356	7158.8232	1332.1468	51248749.5299
2018-01-12 05:00:05	40.705 sec	220	8643.0022	1359.0649	74701487.0318	7157.9702	1332.8968	51236537.8040
2018-01-12 05:00:07	42.398 sec	230	8640.0737	1359.5795	74650872.7583	7157.2868	1333.5512	51226753.8626
2018-01-12 05:00:09	44.097 sec	240	8637.6609	1360.2356	74609185.8697	7156.7269	1334.3417	51218739.5865
2018-01-12 05:00:09	44.097 sec	240	8637.6609	1360.2356	74609185.8697	7156.7269	1334.3417	51218739.5865
2018-01-12 05:00:10	45.796 sec	250	8635.0084	1360.6846	74563370.1252	7156.4190	1334.9750	51214332.4714
2018-01-12 05:00:12	47.505 sec	260	8632.7606	1361.3573	74524556.2771	7155.8910	1335.7781	51206775.7595
2018-01-12 05:00:14	49.219 sec	270	8630.7712	1361.8247	74490211.2320	7155.4054	1336.3291	51199826.1304
2018-01-12 05:00:16	50.914 sec	280	8628.0502	1362.4906	74443250.4486	7155.0931	1337.1434	51195357.4093
2018-01-12 05:00:17	52.606 sec	290	8625.7098	1363.0914	74402870.0040	7154.8208	1337.8344	51191461.1964
2018-01-12 05:00:19	54.296 sec	300	8622.3181	1363.6640	74344368.9489	7154.7706	1338.5444	51190741.9991
2018-01-12 05:00:21	55.981 sec	310	8619.5525	1364.2310	74296684.6338	7154.5613	1339.2101	51187747.2625

2018-01-12 05:00:22	57.674 sec	320	8617.0904	1364.8494	74254247.4339	7154.3276	1339.9225	51184403.9750
2018-01-12 05:00:24	59.351 sec	330	8614.5857	1365.3711	74211086.2003	7154.3239	1340.6075	51184350.4221
2018-01-12 05:00:26	1 min 1.337 sec	340	8611.9411	1365.7668	74165530.0320	7154.1850	1341.1244	51182363.5118
2018-01-12 05:00:28	1 min 2.989 sec	350	8609.3809	1366.2876	74121439.7302	7154.4086	1341.7290	51185561.9251
2018-01-12 05:00:29	1 min 4.671 sec	360	8607.0908	1366.7178	74082011.8457	7154.4398	1342.2724	51186008.1549
2018-01-12 05:00:31	1 min 6.335 sec	370	8604.3107	1367.0999	74034162.3709	7154.4259	1342.7241	51185809.9972
2018-01-12 05:00:33	1 min 8.024 sec	380	8601.9488	1367.5020	73993523.4138	7154.4973	1343.2698	51186831.1735

▾ OUTPUT - TRAINING_METRICS

model	GBM_model_Oracle_1515730253930_3
model_checksum	-1527561827186006016
frame	train.hex
frame_checksum	-793620439018086784
description	·
model_category	Regression
scoring_time	1515733233570
predictions	·
MSE	73993523.413837
RMSE	8601.948815
nobs	1135059
custom_metric_name	·
custom_metric_value	0
r2	0.064342
mean_residual_deviance	73993523.413837
mae	1367.502040
rmsle	NaN

▾ OUTPUT - VALIDATION_METRICS

model	GBM_model_Oracle_1515730253930_3
model_checksum	-1527561827186006016
frame	valid.hex
frame_checksum	1035512935444540288
description	·
model_category	Regression
scoring_time	1515733233582
predictions	·
MSE	51186831.173530
RMSE	7154.497269
nobs	288005
custom_metric_name	·
custom_metric_value	0
r2	0.066322
mean_residual_deviance	51186831.173530
mae	1343.269825
rmsle	NaN

▾ OUTPUT - VARIABLE IMPORTANCES

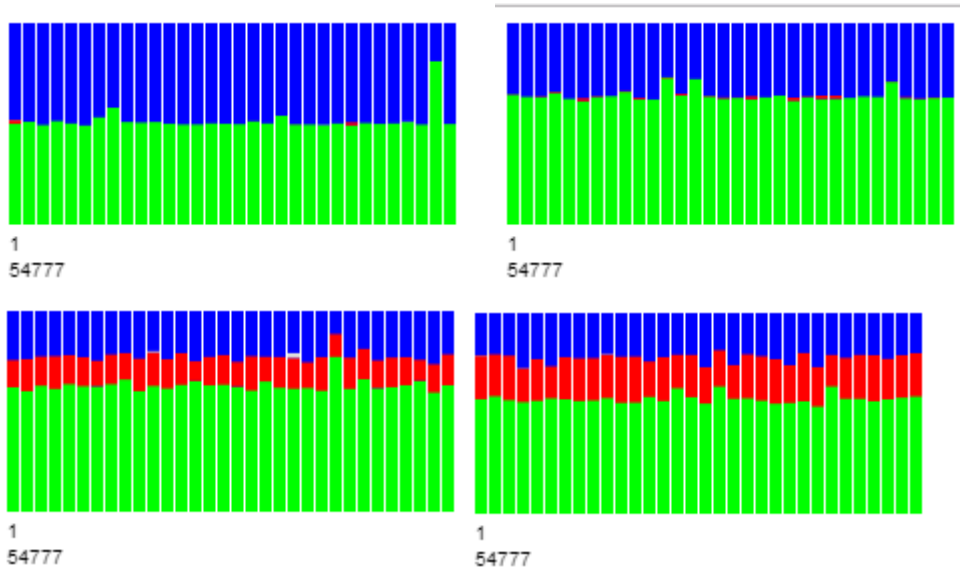
variable	*relative_importance*	*scaled_importance*	*percentage*
COD_SEGMENTO	137188410916864.0	1.0	0.5195
GLOBAL_POLIZAS	23456194232320.0	0.1710	0.0888
PROVINCIA	17103080914944.0	0.1247	0.0648
TOTAL_SINIES_DESDE2005	16851018973184.0	0.1228	0.0638
YEARS_VIGENTE	10464923746304.0	0.0763	0.0396
EDAD	5807036956672.0	0.0423	0.0220
SOCGRUPOMOSAIC	5258364321792.0	0.0383	0.0199
NOTA1	4853859352576.0	0.0354	0.0184
DP300	4172020449280.0	0.0304	0.0158
CONT_VIGOR	3626167697408.0	0.0264	0.0137
SOCRENTAMEDIA	2910918017024.0	0.0212	0.0110
QUARTER_BAJA	2651075379200.0	0.0193	0.0100
NOTA2V	2574363394048.0	0.0188	0.0097
DNIEVESUE	1994996187136.0	0.0145	0.0076
SOCPAROTOT	1843342344192.0	0.0134	0.0070
SOCTRANSITORIEDAD	1723985559552.0	0.0126	0.0065
QUARTER_ALTA	1647618949120.0	0.0120	0.0062
SOCANTMA5	1558238855168.0	0.0114	0.0059
SOCSEGMENTOSEGUROS	1216187727872.0	0.0089	0.0046
CLASIFICACION_EMPLEADO	1210813513728.0	0.0088	0.0046

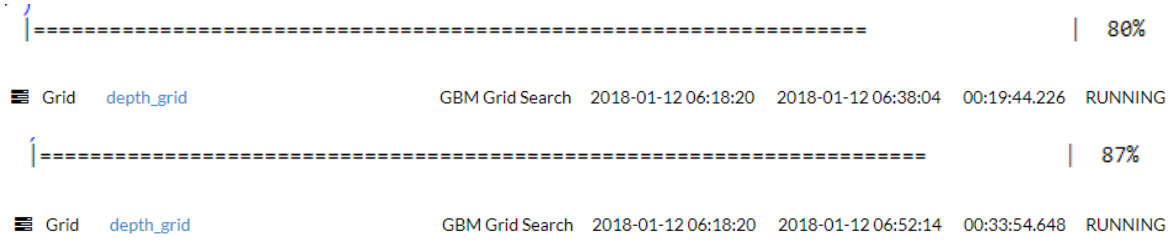
SOCSEGMEDEMP	1103722053632.0	0.0080	0.0042
COMU_AUTONOMA	1078035611648.0	0.0079	0.0041
DIAS_TMAX_30	947804438528.0	0.0069	0.0036
SOCTIPVIV	886084534272.0	0.0065	0.0034
DTORMENTA	849924587520.0	0.0062	0.0032
SOCMOTPRECIO	785230135296.0	0.0057	0.0030
SOCCALIDADVIV	778267918336.0	0.0057	0.0029
SOCVIDMUERVENC	768895221760.0	0.0056	0.0029
SOCMOTRAPI	755648233472.0	0.0055	0.0029
SOCFHOGAR	714781818880.0	0.0052	0.0027
SOCCONTBANCO	712122302464.0	0.0052	0.0027
SOCURBANIDAD	700987080704.0	0.0051	0.0027
R_MAX_VEL	681282699264.0	0.0050	0.0026
DNIEBLA	680128282624.0	0.0050	0.0026
SOCCONTMEDIADOR	503150116864.0	0.0037	0.0019
EXT_EXT	478473027584.0	0.0035	0.0018
SOCFFAMILIAS	441898795008.0	0.0032	0.0017
SOCCONTAUTMEDIADOR	429489061888.0	0.0031	0.0016
SOCCONTAUTINT	301708541952.0	0.0022	0.0011
SOCCOCHES1	273523884032.0	0.0020	0.0010
SOCSEGMEDPARC	268901908480.0	0.0020	0.0010
SOCTEROPC	256298598400.0	0.0019	0.0010
SOCCONTAUTBANCO	247026253824.0	0.0018	0.0009
SOCCONTCIA	181829664768.0	0.0013	0.0007
SOCESTRES	160156024832.0	0.0012	0.0006
SOCMOTNECE	153364299776.0	0.0011	0.0006
HABITANTES	146726682624.0	0.0011	0.0006
SOCMOTOFER	141031653376.0	0.0010	0.0005
DROCIO	134020800512.0	0.0010	0.0005
SOCMOTCONF	131571449856.0	0.0010	0.0005
SOCMOTSERV	123011727360.0	0.0009	0.0005
SOCCONTM5	79979200512.0	0.0006	0.0003
DENSIDAD	23544770560.0	0.0002	0.0001
SOCCOMPRAINTERNET	22782681088.0	0.0002	0.0001
SOCTRCF	7880942592.0	0.0001	0.0

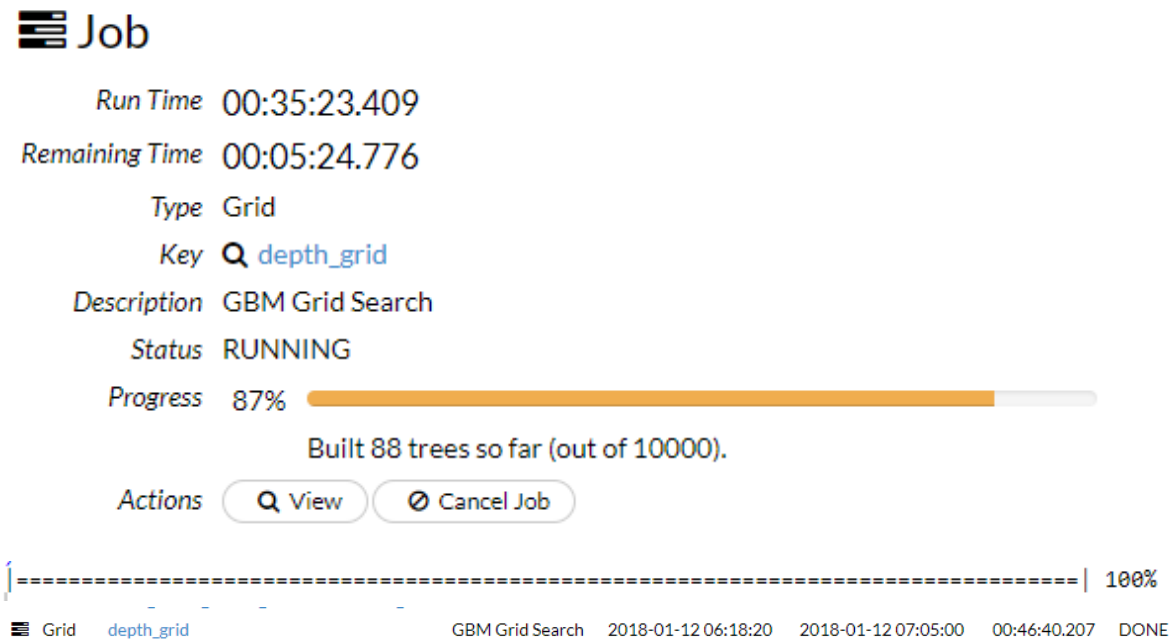
Modelo 4

#CARTERSIAN SEARCH GRID

Status CPU







El major modelo de los 15 creados es el 2

```
> sortedGrid <- h2o.getGrid("depth_grid", sort_by="MSE", decreasing = FALS
E)
> sortedGrid
H2O Grid Details
================

Grid ID: depth_grid
Used hyper parameters:
  -  max_depth
Number of models: 15
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing MSE
   max_depth           model_ids                  mse
1          5  depth_grid_model_2  5.117500716464986E7
2          3  depth_grid_model_1  5.127312511925399E7
3          7  depth_grid_model_3  5.149656945966255E7
4          1  depth_grid_model_0  5.197664570993798E7
5          9  depth_grid_model_4  5.204553613302622E7
```

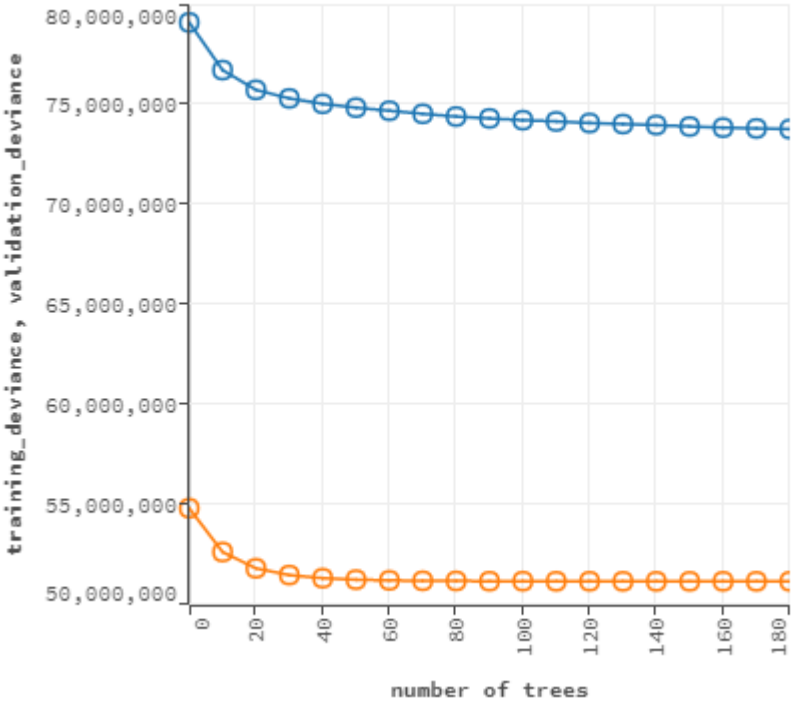
Datos del modelo 2

Model ID: depth_grid_model_2

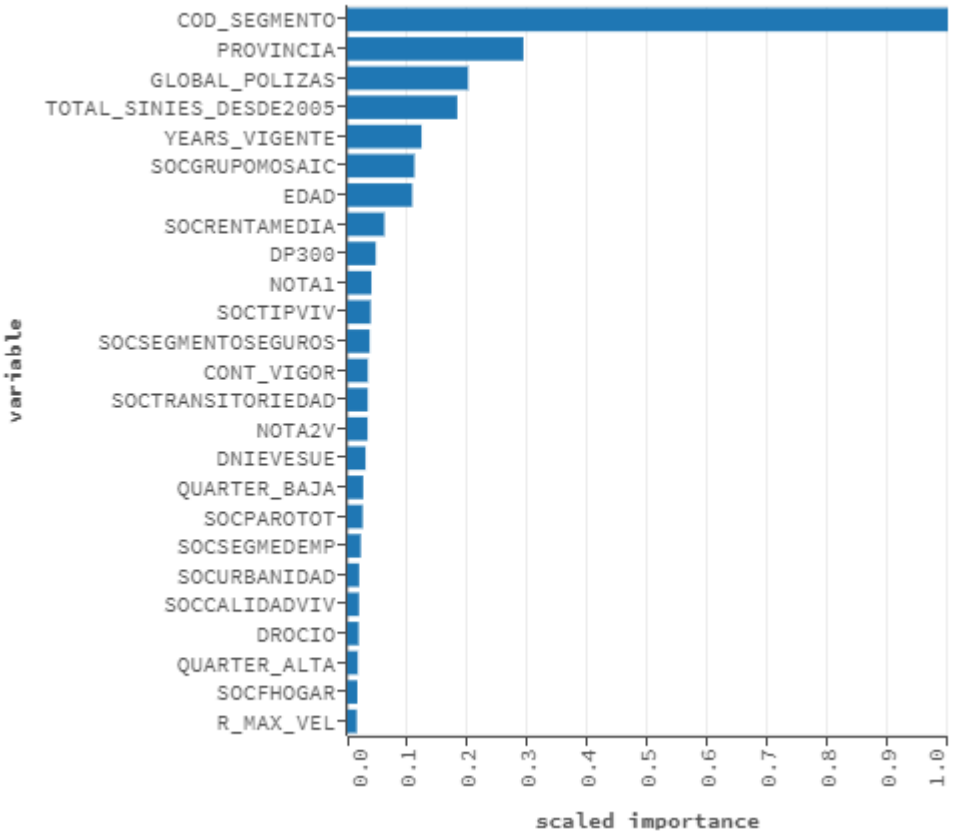
Algorithm: Gradient Boosting Machine

Parameter	*Value*	*Description*
model_id	depth_grid_model_2	Destination id for this model; auto-generated if not specified.
training_frame	train.hex	Id of the training data frame.
validation_frame	valid.hex	Id of the validation data frame.
score_tree_interval	10	Score the model after every so many trees. Disabled if set to 0.
response_column	MARGEN_BRUTO	Response variable column.
ignored_columns		Names of columns to ignore for training.
ntrees	10000	Number of trees.
stopping_rounds	5	Early stopping based on convergence of stopping_metric. Stop if simple moving average of length k of the stopping_metric does not improve for k:=stopping_rounds scoring events (0 to disable)
stopping_metric	MSE	Metric to use for early stopping (AUTO: logloss for classification, deviance for regression)
stopping_tolerance	0.0001	Relative tolerance for metric-based stopping criterion (stop if relative improvement is not at least this much)
max_runtime_secs	1.7976931348623157e+308	Maximum allowed runtime in seconds for model training. Use 0 to disable.
seed	8888	Seed for pseudo random number generator (if applicable)
learn_rate	0.05	Learning rate (from 0.0 to 1.0)
learn_rate_annealing	0.99	Scale the learning rate by this factor after each tree (e.g., 0.99 or 0.999)
distribution	gaussian	Distribution function
sample_rate	0.8	Row sample rate per tree (from 0.0 to 1.0)
col_sample_rate	0.8	Column sample rate (from 0.0 to 1.0)

▾ SCORING HISTORY - DEVIANCE



▾ VARIABLE IMPORTANCES



▾ OUTPUT

cross_validation_models	·
cross_validation_predictions	·
cross_validation_holdout_predictions_frame_id	·
cross_validation_fold_assignment_frame_id	·
model_category	Regression
cross_validation_metrics	·
cross_validation_metrics_summary	·
status	·
start_time	1515734368284
end_time	1515734402287
run_time	34003
init_f	926.992089

▾ OUTPUT - MODEL SUMMARY

number_of_trees	180
number_of_internal_trees	180
model_size_in_bytes	78073
min_depth	5
max_depth	5
mean_depth	5.0
min_leaves	17
max_leaves	32
mean_leaves	26.4333

▾ OUTPUT - SCORING HISTORY

timestamp	duration	number_of_trees	training_rmse	training_mae	training_deviance	validation_rmse	validation_mae	validation_deviance
2018-01-12 05:19:28	1 min 7.980 sec	0	8892.7951	1775.7939	79081803.9061	7404.2503	1742.7509	54822922.7306
2018-01-12 05:19:29	1 min 9.434 sec	10	8758.0050	1467.8227	76702652.0409	7255.1551	1437.7540	52637275.0687
2018-01-12 05:19:31	1 min 11.174 sec	20	8701.4924	1381.7788	75715969.8461	7198.8822	1353.3636	51823905.4806
2018-01-12 05:19:33	1 min 12.908 sec	30	8676.8929	1360.7510	75288470.2883	7175.3138	1333.1512	51485127.9656
2018-01-12 05:19:34	1 min 14.603 sec	40	8661.4017	1357.2036	75019879.8015	7164.5190	1330.3829	51330332.8495
2018-01-12 05:19:36	1 min 16.294 sec	50	8650.2023	1358.2912	74825999.7170	7159.3393	1331.9198	51256138.5965
2018-01-12 05:19:38	1 min 18.397 sec	60	8641.3563	1360.2566	74673038.7423	7156.8339	1334.4840	51220271.4981

2018-01-12 05:19:41	1 min 20.992 sec	70	8632.4648	1362.2379	74519448.9145	7155.3869	1336.7417	51199561.7995
2018-01-12 05:19:43	1 min 22.702 sec	80	8624.5700	1363.9264	74383206.8432	7154.9763	1338.6623	51193685.2752
2018-01-12 05:19:44	1 min 24.571 sec	90	8619.1938	1365.1253	74290502.5941	7154.0257	1340.0550	51180084.1158
2018-01-12 05:19:46	1 min 26.489 sec	100	8614.0247	1365.4759	74201421.0209	7154.1271	1340.6538	51181534.4108
2018-01-12 05:19:48	1 min 28.440 sec	110	8610.3655	1366.3939	74138394.8696	7153.7569	1341.7648	51176237.6067
2018-01-12 05:19:50	1 min 30.310 sec	120	8605.8300	1366.9767	74060309.8185	7153.8748	1342.5858	51177924.9868
2018-01-12 05:19:52	1 min 32.099 sec	130	8602.8345	1367.7379	74008761.4363	7153.5836	1343.3953	51173758.0007
2018-01-12 05:19:54	1 min 33.869 sec	140	8599.8560	1368.2337	73957522.4107	7153.5237	1344.0024	51172901.6508
2018-01-12 05:19:54	1 min 33.869 sec	140	8599.8560	1368.2337	73957522.4107	7153.5237	1344.0024	51172901.6508
2018-01-12 05:19:55	1 min 35.669 sec	150	8595.6337	1368.6033	73884919.0724	7153.5921	1344.5499	51173879.3814
2018-01-12 05:19:57	1 min 37.349 sec	160	8592.3975	1369.0309	73829294.9612	7153.6620	1345.1169	51174880.4013
2018-01-12 05:19:59	1 min 39.191 sec	170	8590.0731	1369.3596	73789356.3573	7153.5962	1345.5439	51173938.6613
2018-01-12 05:20:01	1 min 41.603 sec	180	8588.2758	1369.6455	73758481.4257	7153.6709	1345.9270	51175007.1646

▾ OUTPUT - TRAINING_METRICS

model	depth_grid_model_2
model_checksum	-2275336255467814400
frame	train.hex
frame_checksum	-793620439018086784
description	·
model_category	Regression
scoring_time	1515734402273
predictions	·
MSE	73758481.425716
RMSE	8588.275812
nobs	1135059
custom_metric_name	·
custom_metric_value	0
r2	0.067314
mean_residual_deviance	73758481.425716
mae	1369.645460
rmsle	NaN

▾ OUTPUT - VALIDATION_METRICS

model	depth_grid_model_2
model_checksum	-2275336255467814400
frame	valid.hex
frame_checksum	1035512935444540288
description	·
model_category	Regression
scoring_time	1515734402285
predictions	·
MSE	51175007.164650
RMSE	7153.670887
nobs	288005
custom_metric_name	·
custom_metric_value	0
r2	0.066537
mean_residual_deviance	51175007.164650
mae	1345.926990
rmsle	NaN

▾ OUTPUT - VARIABLE IMPORTANCES

variable	*relative_importance*	*scaled_importance*	*percentage*
COD_SEGMENTO	30840169955328.0	1.0	0.3647
PROVINCIA	9037931347968.0	0.2931	0.1069
GLOBAL_POLIZAS	6201043582976.0	0.2011	0.0733
TOTAL_SINIES_DESDE2005	5643636310016.0	0.1830	0.0667
YEARS_VIGENTE	3809312505856.0	0.1235	0.0450
SOCGRUPOMOSAIC	3443492913152.0	0.1117	0.0407
EDAD	3329904345088.0	0.1080	0.0394
SOCRENTAMEDIA	1901833093120.0	0.0617	0.0225
DP300	1457619861504.0	0.0473	0.0172
NOTA1	1235621117952.0	0.0401	0.0146
SOCTIPVIV	1195890180096.0	0.0388	0.0141
SOCSEGMENTOSEGUROS	1144149639168.0	0.0371	0.0135
CONT_VIGOR	1051258126336.0	0.0341	0.0124
SOCTRANSITORIEDAD	1046250782720.0	0.0339	0.0124
NOTA2V	1037792116736.0	0.0337	0.0123
DNIEVESUE	932528193536.0	0.0302	0.0110
QUARTER_BAJA	812362170368.0	0.0263	0.0096
SOCPAROTOT	786678808576.0	0.0255	0.0093
SOCSEGMEDEMP	686341554176.0	0.0223	0.0081
SOCURBANIDAD	603472723968.0	0.0196	0.0071

SOCCALIDADVIV	595357532160.0	0.0193	0.0070
DROCIO	577146388480.0	0.0187	0.0068
QUARTER_ALTA	533119631360.0	0.0173	0.0063
SOCFHOGAR	513453588480.0	0.0166	0.0061
R_MAX_VEL	482144518144.0	0.0156	0.0057
SOCMOTPRECIO	449370947584.0	0.0146	0.0053
DTORMENTA	436837679104.0	0.0142	0.0052
SOCCONTBANCO	435981287424.0	0.0141	0.0052
COMU_AUTONOMA	415022710784.0	0.0135	0.0049
SOCMOTRAPI	414980145152.0	0.0135	0.0049
DNIEBLA	385458536448.0	0.0125	0.0046
DIAS_TMAX_30	330864295936.0	0.0107	0.0039
CLASIFICACION_EMPLEADO	327954497536.0	0.0106	0.0039
SOCCONTMEDIADOR	321013186560.0	0.0104	0.0038
EXT_EXT	291682942976.0	0.0095	0.0034
SOCANTMA5	285138911232.0	0.0092	0.0034
SOCFFAMILIAS	239788752896.0	0.0078	0.0028
SOCTEROPC	186512932864.0	0.0060	0.0022
SOCVIDMUERVENC	165733007360.0	0.0054	0.0020
SOCCONTCIA	127576383488.0	0.0041	0.0015
SOCMOTOFER	118923698176.0	0.0039	0.0014
SOCCOCHES1	104636284928.0	0.0034	0.0012
SOCTRCF	91158740992.0	0.0030	0.0011
HABITANTES	79209201664.0	0.0026	0.0009
SOCCONTM5	69311225856.0	0.0022	0.0008
DENSIDAD	68170772480.0	0.0022	0.0008
SOCMOTCONF	67024998400.0	0.0022	0.0008
SOCCONTAUTINT	63168208896.0	0.0020	0.0007
SOCCONTAUTBANCO	61222858752.0	0.0020	0.0007
SOCMOTSERV	46854062080.0	0.0015	0.0006
SOCMOTNECE	39742095360.0	0.0013	0.0005
SOCSEGMEDPARC	14568518656.0	0.0005	0.0002
SOCCOMPRAINTERNET	11993580544.0	0.0004	0.0001
SOCESTRES	10445992960.0	0.0003	0.0001
SOCCONTAUTMEDIADOR	8294044160.0	0.0003	0.0001