Guía práctica de Microsoft Azure Machine Learning

CONTENIDO

1. INTRODUCCIÓN

Azure ML es un servicio que está basado en la plataforma "Microsoft Azure". Azure ML, al igual que el resto de las aplicaciones de "Microsoft Azure", está basado en la nube, por lo que no es necesario ningún tipo de infraestructura previa para llevar a cabo proyectos. Además, es un servicio cien por cien plataforma de servicio (PaaS), esto hace que herede ciertas ventajas de muchos otros servicios de plataforma. Algunas de ellas son: Rápido aprovisionamiento, alta disponibilidad, escalabilidad, manejo de la infraestructura, etc.

En Stratebi, como Partners Certificados en Microsoft y especialistas en Analytics, os vamos a contar los principales puntos.

Además, Azure ML se utiliza para el desarrollo de modelos de Machine Learning, desde el entrenamiento del modelo, hasta su implementación y automatización de este. También es posible realizar un seguimiento de los modelos de Machine Learning que se hayan diseñado e implementado.

Esta plataforma está diseñada para ser usada en la implementación de diferentes tipos de algoritmos de Machine Learning, tanto de Supervised Machine Learning, Unsupervised Machine Learning como en Deep Learning, etc.

Asimismo, Azure ML ofrece distintas soluciones para diferentes flujos de trabajo de Machine Learning. Algunas de ellas se presentan a continuación:

- *El diseñador de Azure ML (versión preliminar):* se trata de módulos que te permiten implementar modelos de Machine Learning sin utilizar una sola línea de código, utilizando el método "arrastrar y colocar".
- *Cuadernos de Jupyter Notebook:* Azure ML te permite utilizar tus propios scripts de código utilizando los SDK para Python y R. También puedes usar los cuadernos de ejemplo que puedes encontrar en la plataforma.
- *Extensión de Visual Studio Code.*
- *CLI de Machine Learning.*
- *También puedes utilizar diferentes plataformas Open Source como PyTorch, TensorFlow, Scikit-Learn, etc.*

Por otra parte, Azure ML ha sido desarrollada para que pueda ser utilizada tanto por principiantes en el mundo de la ciencia de datos, como por profesionales del sector. Como se ha dicho en el apartado anterior, Azure ML tiene diferentes opciones a la hora de empezar a

trabajar en su plataforma. Una de ellas es hacer uso de cuadernos de Python para el entrenamiento e implementación de modelos de Machine Learning, utilizando los SDK para Python y R.

Otra opción es usar el diseñador de tipo "arrastrar y colocar" para crear e implementar los modelos de Machine Learning. También existe otra alternativa que consiste en hacer uso del Machine Learning Automatizado para el desarrollo de modelos de Machine Learning.

En este documento se va a explicar paso a paso cómo utilizar la plataforma de Azure ML, resolviendo un caso de uso real. Se va a presentar un problema de clasificación, y se analizará cómo sería implementar haciendo uso de la plataforma de Azure ML, una solución a un problema real para predecir el abandono de los clientes de un banco, en función de sus características. Esta demostración se hará utilizando los tres métodos descritos en el párrafo anterior.

2. CONTEXTUALIZACIÓN

Hoy en día, el mundo genera una cantidad ingente de datos diariamente, de ahí la aparición del término "**Big Data**". Estos datos son una fuente de información para poder predecir comportamientos de la sociedad o población de la que obtenemos las muestras.

Debido a ello, cada vez es más necesario herramientas potentes capaces de trabajar con volúmenes gigantescos de información. Un requisito indispensable para la realización de un proyecto de Big Data con éxito es que los usuarios puedan interactuar con la información, a través de una interfaz sencilla. Con esta idea surgió el concepto de "**nube**", como solución para el almacenamiento de esa enorme cantidad de datos generada diariamente en nuestra sociedad.

La "**nube o cloud**", no es más que un término utilizado para describir una red enorme de servidores remotos de todo el mundo que están interconectados para funcionar como un único ecosistema. Básicamente, en lugar de acceder a archivos y datos de manera local, accede a ellos desde cualquier dispositivo con conexión a internet. Esto fue un gran avance, ya que ahora se puede disponer de la información independientemente de donde se esté, siempre y cuando se tenga conexión a internet. En este punto, quizás lo más relevante sea destacar que al contar con diferentes protocolos de seguridad y posibilidad de almacenar copias de seguridad un número de veces ilimitado, la nube es una buena opción como solución Big Data para un proyecto empresarial.

Debido a la enorme cantidad de información que se produce al día, era fundamental encontrar una solución que fuese capaz de gestionarla, y además extraer de ella conclusiones para mejorar el sistema productivo de las empresas y del mundo en general. Es entonces cuando aparecen máquinas que son capaces de hacer predicciones precisas sin que necesariamente estén programadas para ello. Y es precisamente esa capacidad que tienen las máquinas de aprender por sí mismas a partir de un conjunto de datos dado a lo que se llama "**Aprendizaje Automático o Machine Learning**". Además, las máquinas son capaces también de cambiar y ajustar los distintos algoritmos mientras que procesan información y conocen el entorno.

En el apartado anterior, se ha definido Azure ML como un servicio cien por cien plataforma de servicio (**PaaS**). Cuando se habla de plataforma como servicio (**PaaS**), plataforma de aplicación como servicio (**aPaaS**) o servicio basado en la plataforma, se habla de una categoría de servicios de computación en la nube que proporciona una plataforma que

permite a los clientes desarrollar, ejecutar y administrar aplicaciones sin la complejidad de construir y mantener una infraestructura típicamente asociada con el desarrollo y lanzamiento de una aplicación.

3. AZURE MACHINE LEARNING

En primer lugar, es importante conocer cuáles son las empresas pioneras en proporcionar servicios de Big Data en la nube. Estas empresas son: Amazon Web Services (AWS), Google Cloud Platform (GCP) y Microsoft Azure.

Como se ha mencionado en el apartado anterior, implementar una solución basada en la nube, aparte de ser más segura, permite a las empresas ahorrar en costes, ya que teniendo acceso a una de las tres plataformas anteriores que proporcionan servicios en el cloud, y configurando el entorno de desarrollo en el que se va a trabajar, es posible tener acceso a la infraestructura necesaria para empezar a trabajar en un proyecto, sin costes adicionales, más que los del tipo de subscripción que se desee, en función de la necesidad que se quiera cubrir, o el proyecto que se quiera abordar. Del tipo de subscripciones que existen en Azure ML se hablará más adelante.

Es importante recalcar que el concepto de servicios en la nube, aunque en términos meramente teóricos, sea aplicado a nivel empresarial, también pueden ser utilizados por cualquier usuario que quiera disfrutar de las múltiples funcionalidades de estos servicios. Esto es gracias a que las tres empresas mencionadas antes, ofrecen diferentes opciones en el uso de sus servicios. Normalmente se puede disponer de unos servicios con una cobertura mínima de manera totalmente gratuita. La cuota que se paga por estos servicios varía mucho en función de la cantidad, potencia que se demande, número de servidores virtuales que se quieran utilizar, etc. En los tres casos no existen costes iniciales ni cargos por cancelación, y además se paga solamente por el uso de los servicios que se utilicen.

A continuación, se enumeran algunas razones para usar Microsoft Azure como empresa proveedora de tus servicios en la nube:

- Azure tiene unas restricciones más estrictas que AWS o GCP, por lo que contarás con mayor protección en la nube.
- Debido a sus numerosas soluciones vanguardistas en el sector del cloud, es la plataforma líder por excelencia en proveer este tipo de servicios. Es la empresa que cuenta con el mayor número de regiones con servicios en la nube.
- Azure permite la integración de sus servicios con otras soluciones, herramientas o lenguajes de programación de código abierto, etc.

- Debido a que Microsoft Azure se basa en el GPU para los procesos de sus servicios, posee una alta capacidad de cómputo de proceso, lo que te permite agilizar el aprendizaje de las máquinas, hacer diferentes simulaciones, hacer análisis de datos en tiempo real, etc., todo ello con un alto rendimiento.
- Tiene soluciones totalmente enfocadas a la investigación y análisis de negocios, tales como soluciones implementadas ya para la previsión de la demanda, optimización de inventarios, etc.
- Cuenta con una herramienta propia Azure Cost Management, con el cual puedes controlar la asignación de costos, mediante la optimización del gasto, te permite también la elección del tamaño de las máquinas virtuales que vayas a utilizar en tú proyecto, así como ir visualizando los resultados que se van obteniendo.

Todas las empresas mencionadas anteriormente, ofrecen diferentes servicios en el cloud, que consisten básicamente en automatizar ciertos procesos dentro de un proyecto, buscando optimizar costes, recursos y tiempo.

El debate al que se enfrenta cualquier empresa que quiera hacer uso de estas soluciones de Machine Learning en la nube está clara: Qué procesos automatizar y cuáles no. En la mayoría de los casos casi nunca se automatiza el proceso entero. Muchas veces el miedo a la innovación y/o a lo desconocido, conlleva el rechazo de este tipo de soluciones que te automatizan todas las fases de un proyecto de Machine Learning. Normalmente esto es debido al desconocimiento respecto a uso y alcance de estas soluciones. Muchas veces se prefiere realizar un proyecto por la vía tradicional, y se termina por utilizar los recursos que se sabe que funcionan, como puede en este caso, un equipo de científicos de datos.

Y es verdad que la automatización completa de procesos de Machine Learning no se da en casi ninguna empresa hoy en día. La única empresa dedicada a procesos de Automatización de Machine Learning al completo es DataRobot que fue fundada en 2012. También hay que destacar que cada vez más, las empresas están apostando por la robotización y automatización de procesos, consiguiendo con ello mejorar su productividad y reducir costes, entre sus numerosas ventajas.

En este documento solo se realizará la demostración del funcionamiento de Azure ML, la cual no solo se integra perfectamente con otros servicios de la plataforma de Microsoft Azure, sino que también lo hace con distintas herramientas "**open source**" (Git, MLFlow, etc).

Para concluir con este apartado se hablará de las diferencias existentes entre Azure ML Studio (clásico) y Azure ML. También se comentarán los diferentes tipos y opciones de subscripción que existen, y finalmente se comentará la proyección a futuro que tiene esta plataforma como servicio, así como próximas releases.

En primer lugar, solo existe una gran diferencia entre Azure ML Studio (clásico) y Azure ML, y esta es que mientras Azure ML provee tanto de la opción de usar los SDK para Python y R, además del diseñador d tipo "arrastrar y colocar", Azure ML Studio (clásico), solo cuenta con esta segunda opción. Por ello, es recomendable que los nuevos usuarios utilicen Azure ML, al ser una opción más completa y vanguardista.

En segundo lugar, en cuanto a los diferentes tipos de subscripción existentes, Azure ML ofrece dos opciones en función de la necesidad de Machine Learning que se desee cubrir:

- Basic (disponibilidad general).
- Enterprise (versión preliminar).

En función de la opción elegida, se tendrán unas herramientas de Machine Learning u otras.

Mientras que en el caso de la opción Basic, en la que solo se paga por los recursos utilizados en el proceso de Machine Learning, en la edición Enterprise, se cobrará solamente el consumo de Azure, siempre y cuando se tenga la versión preliminar activa. Puede consultar con más detalle los precios en esta página web: https://azure.microsoft.com/en-us/pricing/details/machine-learning/.

El tipo de opción se tiene que asignar cada vez que se cree un área de trabajo. Y es posible actualizar el tipo de opción que se haya elegido para su área de trabajo. Para más información sobre cómo hacerlo: https://docs.microsoft.com/es-es/azure/machine-learning/how-to-manage-workspace#upgrade.

Cabe destacar que la edición Enterprise tiene consigo ventajas respecto a la opción Basic, tales como la existencia de características mejoradas, la posibilidad de implementar modelos de Machine Learning con poco código, y un nivel de seguridad mayor.

Por último, la última release de Azure ML se realizó el 11/03/2020, y en ella se incorporaron nuevas funcionalidades, y mejoras y correcciones de errores, de algunas de sus características anteriores. Entre ellas destacan:

- Esta versión va a ser la última versión compatible con Python 2.7.

- Versionamiento semántico 2.0.0: Todas las versiones a partir de esta versión tendrán un esquema de numeración nuevo, así como un contrato de Versionamiento Semántico.

4. CONCEPTOS Y TERMINOLOGÍA

En esta parte, se va a introducir solamente aquella terminología que se ha considerado como fundamental para poder entender las diferentes demostraciones que se van a realizar utilizando Azure ML. Para ampliar los conceptos, puede consultar esta página web: https://docs.microsoft.com/es-es/azure/machine-learning/concept-azure-machine-learning-architecture. También se han incluido definiciones de las principales métricas de clasificación, que a su vez son las más utilizadas en el campo de la Estadística y Machine Learning, para cuantificar y medir cómo de bueno es un modelo de clasificación.

En apartados anteriores se ha introducido el término de SDK para Python. El **SDK de Azure para Python** es un conjunto de bibliotecas que le permiten trabajar en Azure para sus necesidades de administración, tiempo de ejecución o datos.

Cuando se habla de **Actividades**, esto no es más que una operación que tarda en ejecutarse un período de tiempo largo. Las actividades permiten la supervisión de todo el proceso de las operaciones que se realicen a través del SDK o de la interfaz de usuario web.

Un término importante es el de **Área de trabajo**, el cual es un punto central que permite interactuar y trabajar con los distintos objetos que cree al realizar su proyecto de Machine Learning. También es posible compartir un área de trabajo con diferentes usuarios de su organización, etc.

Un **Experimento** es un conjunto de ejecuciones de un script en concreto. Un experimento siempre forma parte de un área de trabajo. Es importante poner un nombre al experimento que se esté realizando al enviar una ejecución. Si esto no se hace se crea de manera automática un experimento nuevo con ese nombre. Cuando se realiza un experimento, toda la información relativa a la ejecución que se realice se guarda en él.

Una **Ejecución** es una única ejecución de un script de entrenamiento. Varias ejecuciones dan lugar a un experimento. Azure ML deja registradas todas las ejecuciones que se hagan, y además almacena información sobre el experimento (Marcas de tiempo y duración, diferentes métricas del script, archivos de salida, directorio donde se encuentran los scripts antes de la ejecución, etc.)

Cuando se realiza un envío de un script para el entrenamiento de un modelo de Machine Learning, se produce una ejecución. Una ejecución puede estar formada por un número de ejecuciones ilimitado.

Cuando se habla de cómo se debe de ejecutar un script en un destino predefinido a través de una serie de instrucciones, aparece el concepto de **Configuración de ejecución**. Una configuración de ejecución puede ser guardada junto con el script de entrenamiento en un archivo del directorio, o usarse en el envío de la ejecución mediante la creación de un objeto en la memoria.

Cuando se quiere crear o administrar un flujo de trabajo dentro de un proceso de Machine Learning se utilizan las **Canalizaciones de Machine Learning**. Una canalización puede estar formada por distintas fases, y cada fase puede estar a su vez compuesta por diferentes pasos. Estos a su vez pueden ser ejecutados en varios destinos de proceso. También pueden ser ejecutados sin necesidad de volver a ejecutar los pasos anteriores, si estos no han sido modificados. Las canalizaciones pueden ser reutilizadas si los datos no han sido cambiados.

Probablemente uno de los términos más importantes en Azure ML sea el de **Modelos**. Básicamente un modelo es un script de código que tiene una entrada y una salida. Azure ML tiene una serie de algoritmos ya predefinidos y preparados para crear un modelo de Machine Learning. Cuando se habla de **Entrenamiento**, esto es un proceso que de manera repetitiva genera un modelo entrenado, el cual tiene un proceso de aprendizaje que le permite absorber toda la información recibida en ese proceso de aprendizaje durante el proceso de aprendizaje.

Cuando se realiza una ejecución se crea de manera automática un modelo. También es posible utilizar un modelo que haya sido ya previamente entrenado fuera del entorno de Azure ML. Un modelo puede ser registrado en un área de trabajo. Cuando se crea un modelo de Machine Learning, se puede elegir de entre diferentes bibliotecas open source (Scikit-Learn, XGBoost, PyTorch, TensorFlow, etc.), el que se prefiera.

Cada modelo se identifica por un nombre y una versión. Cuando se produce un registro con un mismo nombre, Azure ML interpreta que es una versión nueva, y esta se incrementa y se registra con el nombre que se le haya dado. El **Registro de modelos** contiene un seguimiento de todos los modelos existentes en el área de trabajo. Cuando se haya registrado el modelo, se le puede dar una etiqueta de metadatos de manera adicional, para posteriormente poder utilizar esa etiqueta al buscar los modelos. Obviamente no es posible eliminar un modelo registrado si se está haciendo uso de este.

Cuando se habla de **Entornos de Azure ML,** se habla de entidades con control de versiones en el área de trabajo, que hacen que los flujos de trabajo de Machine Learning sean reproducibles, auditables y portátiles en diferentes destinos de proceso. Se utilizan también para definir la configuración (Docker, Python, Spark, etc.) que vaya a ser utilizada a la hora de crear un

entorno de trabajo. Se pueden reutilizar entornos tanto para la parte del entrenamiento del modelo, como en la de implementación.

Como se ha comentado unas líneas más arriba, se pueden utilizar diferentes bibliotecas open source, tales como PyTorch, TensorFlow, Scikit-Learn, etc. Pero no solo eso, sino que además para facilitar el entrenamiento de los modelos haciendo uso de esas bibliotecas, aparece un nuevo término: **Estimator**. Este le permitirá crear de una manera muy sencilla diferentes configuraciones de ejecución. Normalmente se usará un objeto estimator genérico para la parte del entrenamiento del modelo de Machine Learning.

De manera muy simplificada, un **Punto de Conexión** es utilizada en las implementaciones que se hagan de los dispositivos integrados y consiste en crear una instancia del modelo en un servicio web, el cual puede encontrarse en un módulo IoT o en el cloud.

Una **Instancia de proceso** puede ser utilizada como destino de proceso en los trabajos de entrenamiento e inferencia. Consiste en una estación de trabajo administrada por completo en el cloud, y que está formada por distintas herramientas y entornos ya instalados para la creación de modelos de Machine Learning.

Para finalizar la primera parte de los conceptos sobre Azure ML, se define **Destino de proceso** como el lugar donde puede ser ubicado el recurso de proceso (tanto en su equipo local como en el cloud), en el que posteriormente se va a ejecutar el script de entrenamiento o se hospeda la implementación que se haga del servicio web.

A continuación, se introducirán los conceptos de las principales métricas de evaluación de los modelos de clasificación.

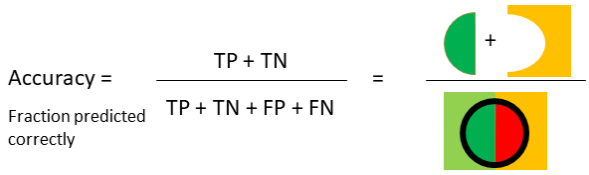
En términos generales, una **Matriz de Confusión** es una herramienta que permite la visualización de cómo de bueno es un algoritmo que se emplea en aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real.

Una de las ventajas de utilizar esta métrica es que, al ser un método tan visual, te permite rápidamente cuantificar cuantas clases estas prediciendo de forma correcta, y cuáles no.

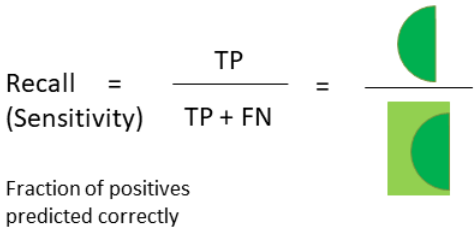
Confusion Matrix		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

- True positive — actual = 1, predicted = 1
- False positive — actual = 0, predicted = 1
- False negative — actual = 1, predicted = 0
- True negative — actual = 0, predicted = 0

La **Precisión o Accuracy** es la relación entre el número de predicciones correctas y el número total de muestras de entrada. Suele funcionar bien solo si hay el mismo número de muestras que pertenecen a cada clase.



Recall o Sensitivity calcula cuántos de los positivos reales captura nuestro modelo al etiquetarlo como positivo (verdadero positivo). Recall será la métrica del modelo que se usará para seleccionar nuestro mejor modelo cuando haya un alto costo asociado con Falso negativo.



F1-score es la media armónica entre precisión y recuperación. El rango para la puntuación F1 es [0, 1]. Le dice qué tan preciso es su clasificador (cuántas instancias clasifica correctamente), así como qué tan robusto es (no pierde un número significativo de instancias).

$$F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * (precision * recall)}{precision + recall}$$

Area Under Curve (AUC) es una de las métricas más utilizadas para la evaluación. Se utiliza para problemas de clasificación binaria. El AUC de un clasificador es igual a la probabilidad de que el clasificador clasifique un ejemplo positivo elegido al azar más alto que un ejemplo negativo elegido al azar. Cuanto mayor sea el área bajo la curva, mejor será nuestro modelo.

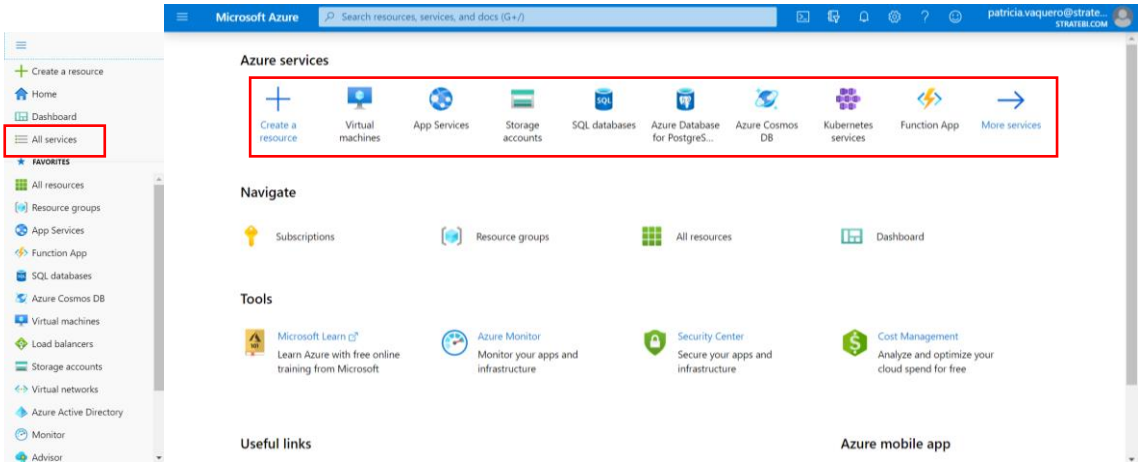
Una **Curva ROC** es una representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación. Otra interpretación de este gráfico es la representación de la razón o ratio de verdaderos positivos (VPR = Razón de Verdaderos Positivos) frente a la razón o ratio de falsos positivos (FPR = Razón de Falsos Positivos) también según se varía el umbral de discriminación (valor a partir del cual decidimos que un caso es un positivo). Cuanto mayor sea el área bajo la curva, mejor será nuestro modelo.

5. ELEMENTOS MICROSOFT AZURE

En este apartado se va a introducir la interfaz del portal de Microsoft Azure.

Una vez que creada una cuenta nueva de Azure, y previamente habiendo completado todo el proceso de registro (registro con el correo profesional (live, Outlook, etc.), validación del número de teléfono introducido, configuración de la tarjeta de crédito, y se hayan aceptado las condiciones de Azure), aparece la siguiente imagen del portal de Microsoft Azure. En ella se pueden ver los diferentes servicios que están disponibles.

En la parte de arriba aparecen diferentes opciones para configurar el portal de Azure. A la izquierda aparece un desplegable, que si se abre contiene los diferentes servicios que tiene Azure dentro de la plataforma. Si pulsa en la opción "**All services**", podrá ver todos los servicios con los que puede trabajar en Azure Microsoft.



6. EJEMPLO 1

Antes de entrar con las demostraciones, me parece relevante incluir en este apartado, el término **Flujo de trabajo**.

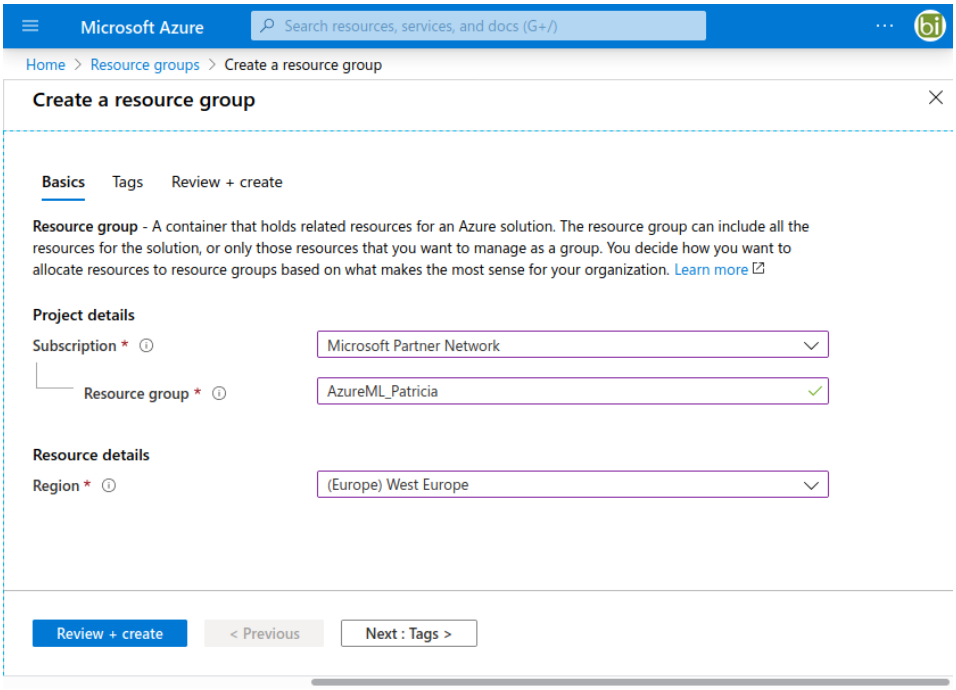
En términos generales, cualquier modelo de Machine Learning está compuesto por un flujo de trabajo similar, consistente en los siguientes pasos:

1. **Entrenamiento del modelo**
- Se crea el modelo mediante scripts en Python o en R, o bien mediante la opción del diseñador visual.
- Se tiene que configurar el destino de proceso.
- Para que los scripts sean ejecutados en un entorno, debe ser enviados a un destino de proceso previamente configurado en ese entorno. En la parte del entrenamiento de los modelos de Machine Learning, los scripts pueden leer y escribir en almacenes de datos. Cualquier salida generada en el proceso del entrenamiento del modelo, será guardada como una ejecución en su área de trabajo, y además formarán lo que se llaman experimentos.

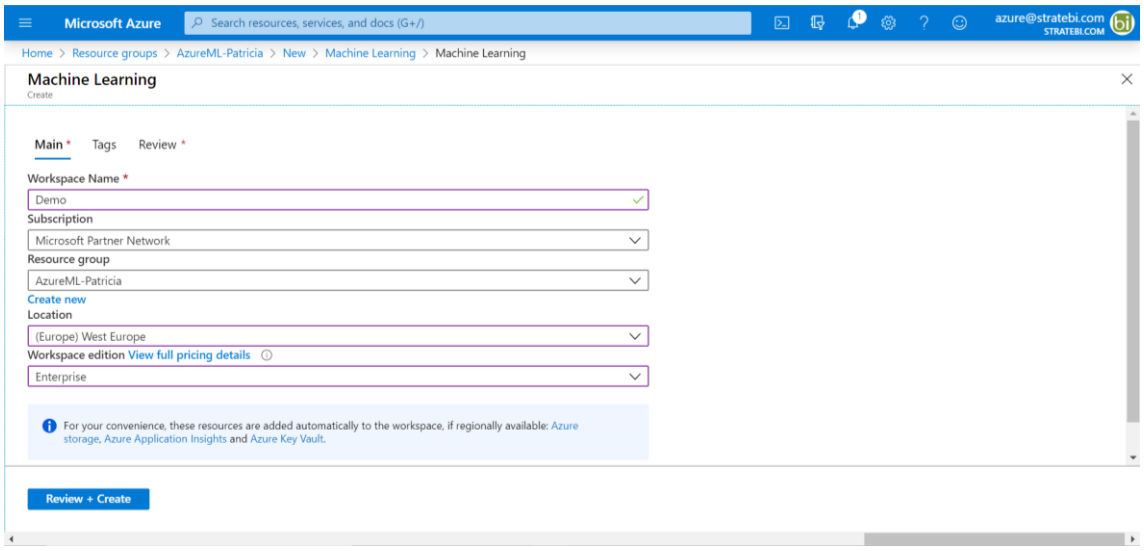
2. **Paquete:** Las ejecuciones realizadas con éxito serán almacenadas en un registro de modelos.
3. **Validación del experimento**: Si los resultados obtenidos no son los esperados, vuelva al paso uno y repita los scripts.
4. **Implementación del modelo:** El modelo desarrollado puede ser implementado, tanto en un servicio Web de Azure, como en un dispositivo IoT Edge.
5. **Supervisión.**

A continuación, una vez realizada esa introducción teórica fundamental para entender lo que se va a realizar en las posteriores demostraciones, se seguirá explicando cómo crear un espacio de trabajo nuevo.

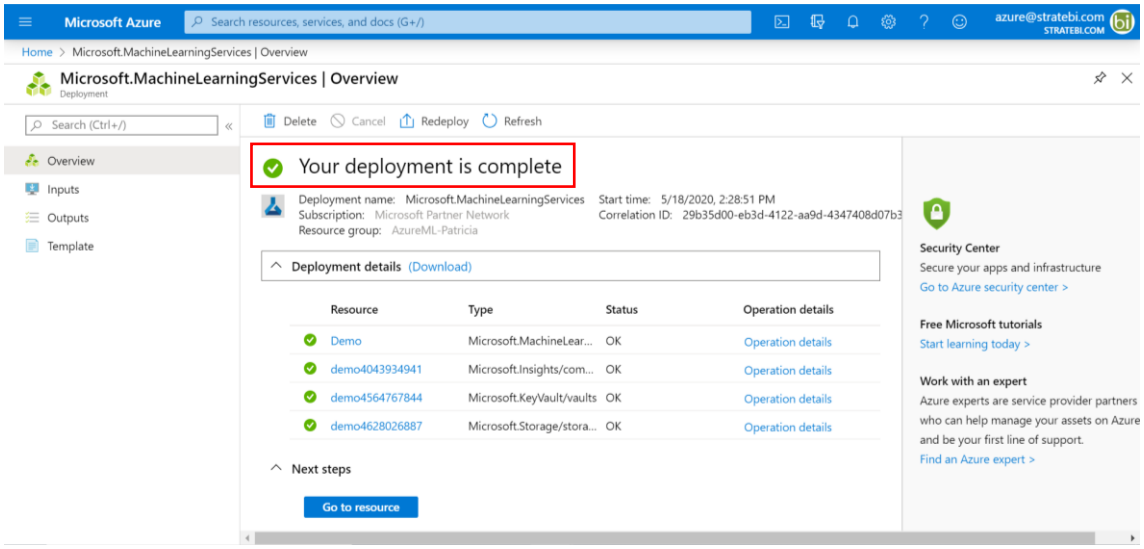
Lo primero que se tiene que hacer antes de nada es crear un "Resource group". Esto es el grupo de recursos con los servicios que se utilizarán para un proyecto. En este caso se le ha llamado "AzureML_Patricia".



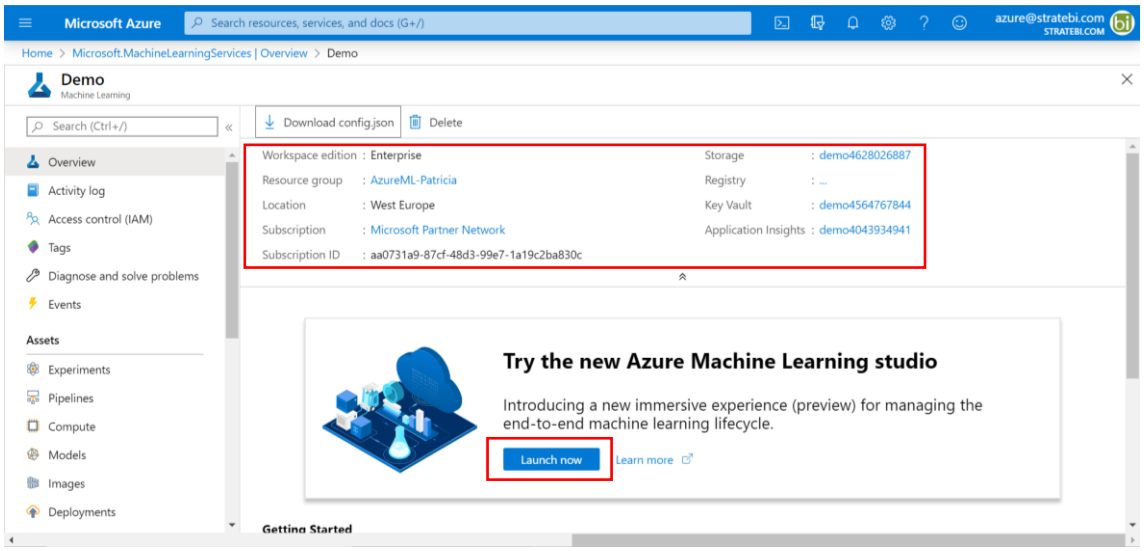
Después se crea un "Workspace" llamado "Demo".



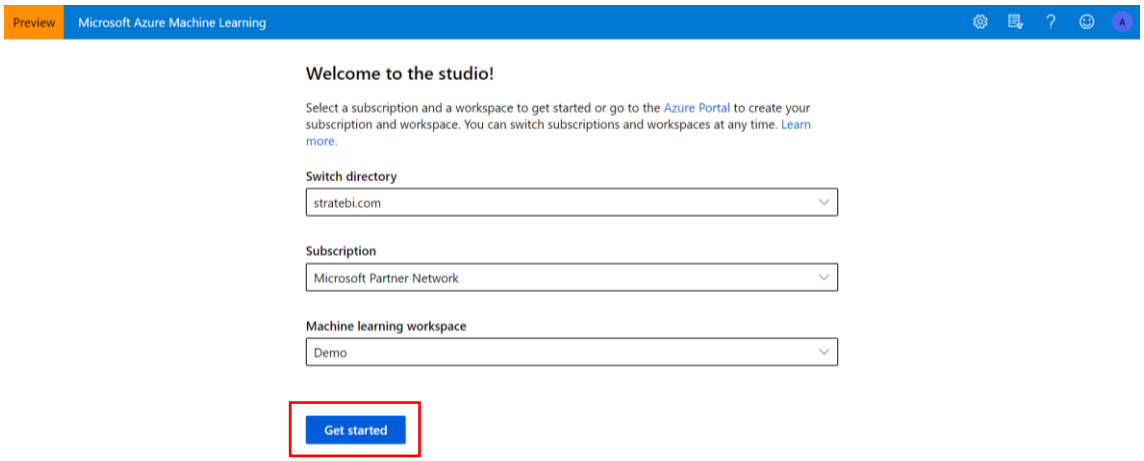
En la siguiente imagen se puede ver que el "Workspace" llamado "Demo" se ha creado con éxito.

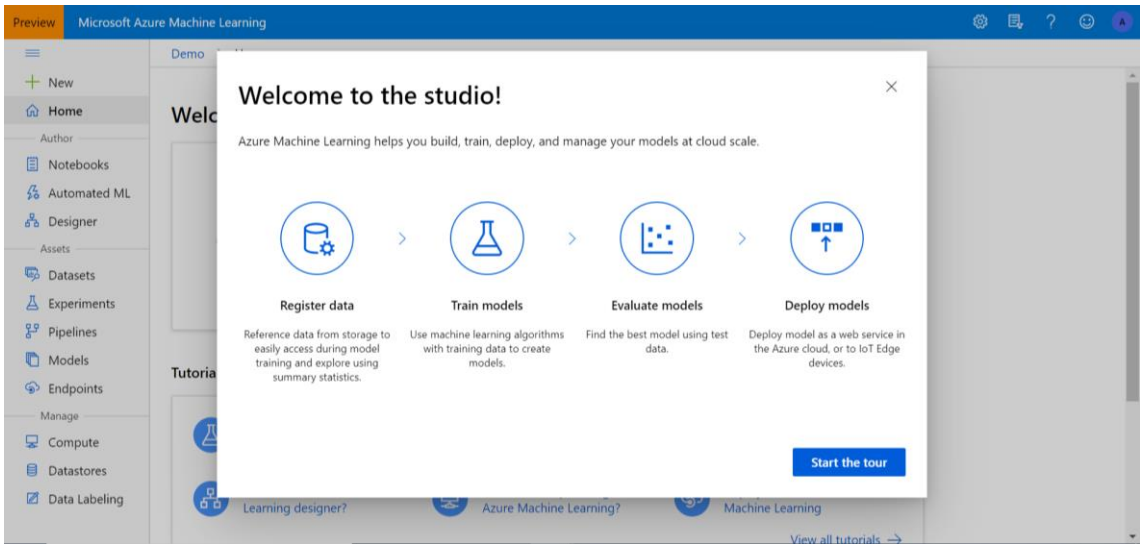


Finalmente, en la siguiente imagen aparecen todos los detalles del "Workspace" en el que se trabajará. Hay que pulsar en "Launch now" para empezar a trabajar en Microsoft Azure Machine Learning.

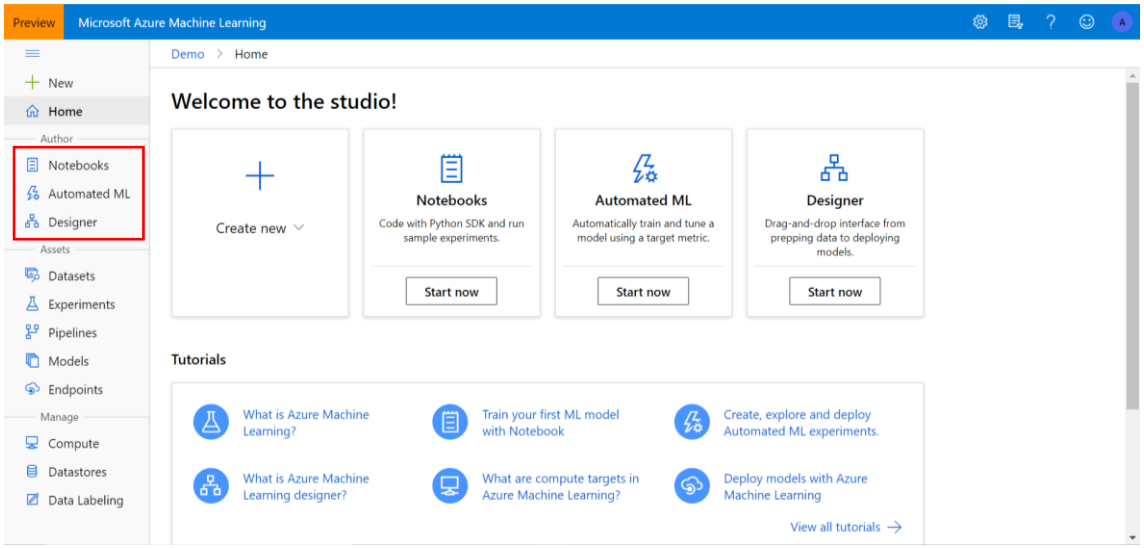


Lo siguiente que aparece es la bienvenida a la interfaz de Azure ML. Se pueden ver tanto la "Switch" del directorio, como el tipo de subscripción que se tiene, el nombre del "Workspace" que ha creado antes, etc. Hay que darle al botón de "Get Started" para continuar.





Azure ML Services Workspace es el punto central de gestión y exploración de experimentos. Al centralizar todos los recursos, sirve como de hub de exploración y monitorización de experimentos y desarrollos. A continuación, como se puede observar en la siguiente imagen, existen diferentes formas de empezar a construir un modelo de Machine Learning (Notebooks, Automated ML y Designer).

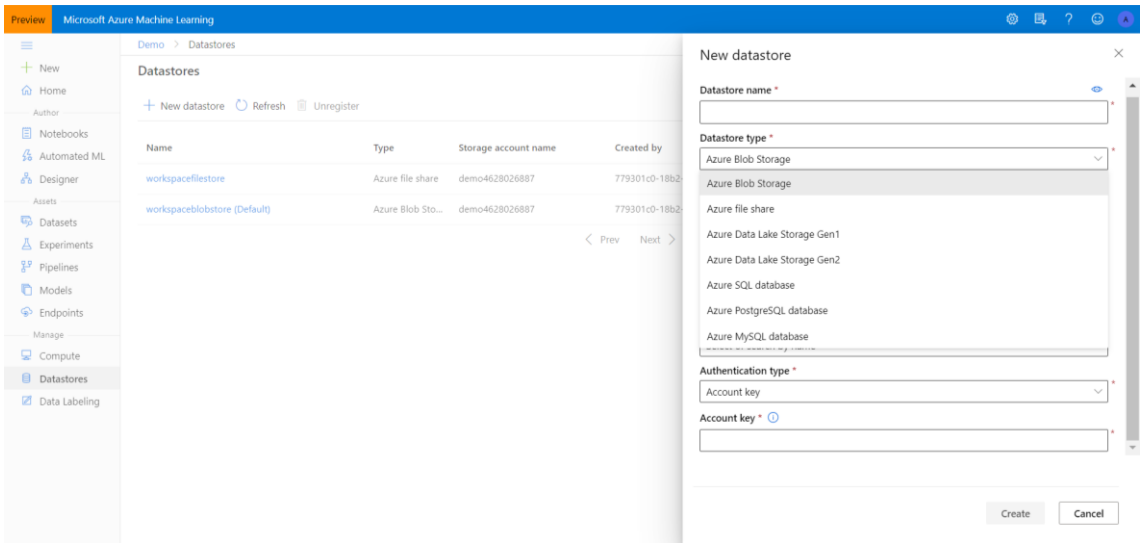


En Azure ML Services se pueden obtener datos de diferentes maneras. Por defecto, el propio servicio tiene dos almacenamientos propios, un Blob Storage y un File Share. El almacenamiento por defecto inicial es el File Share, pero éste puede ser cambiado a través del SDK. Existen diferentes maneras con las que trabajar con los datos, entre ellas están:

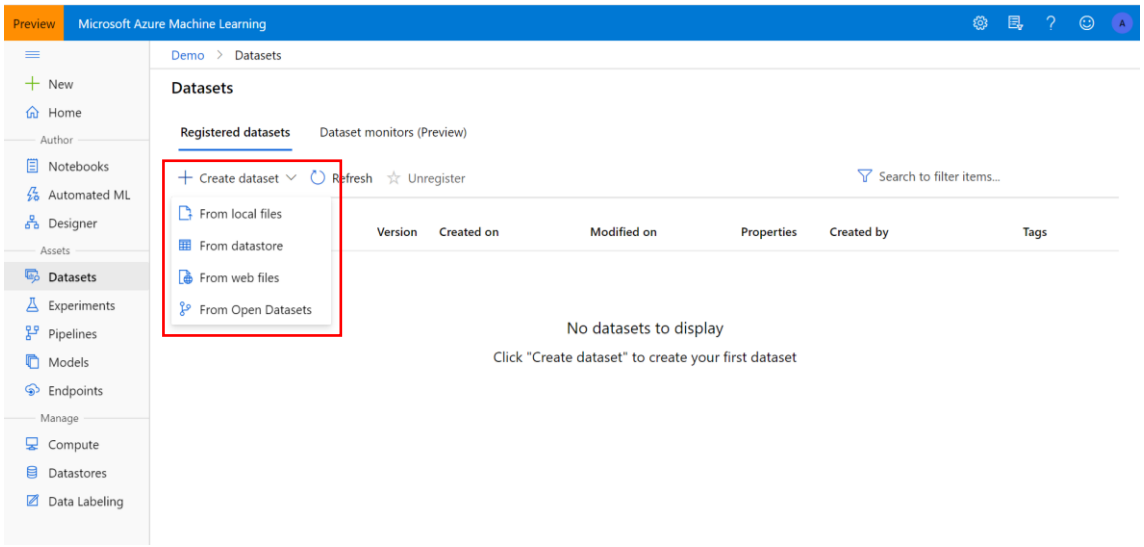
- Se pueden subir los datos a almacenamientos propios que son creados como parte del servicio (los citados Blob Storage y File Share).
- También se pueden registrar los datastores (Azure Blob Storage, Azure File Share, Azure Data Lake Storage Gen1&2, Azure SQL Database, Azure PostgreSQL Database, Azure MySQL Database, etc.), dentro del propio "Workspace" configurando unas credenciales concretas para tener acceso a ellos.

Si la fuente de datos no se puede registrar como Datastore (un bucket S3 de Amazon, por ejemplo), se puede acceder directamente a la fuente desde nuestro script Python. Las credenciales, idóneamente, deberán estar securizadas en un Key Vault.

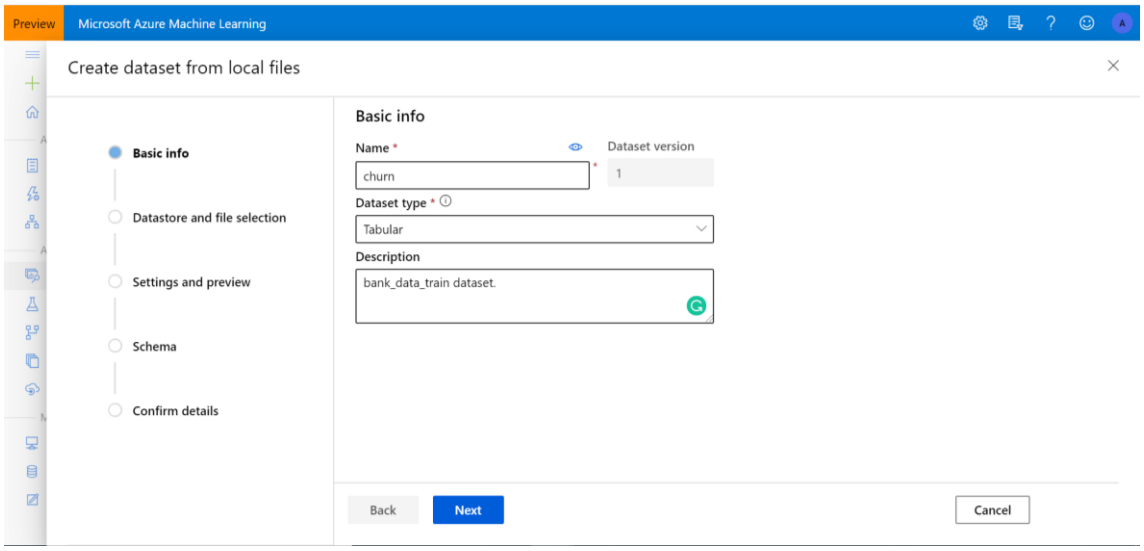
Se pueden crear datastores desde el SDK y desde la interfaz gráfica. La configuración es bastante directa si se tienen los datos de acceso (cuenta y account key o SAS key).



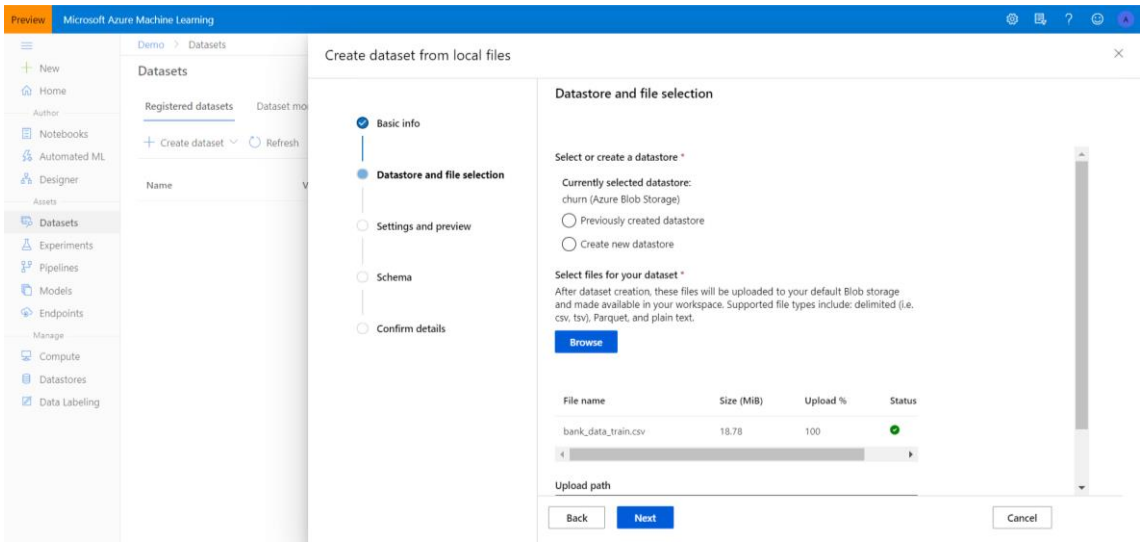
Además, también se pueden crear datasets concretos desde archivos locales, desde un datastore previamente configurado, desde archivos web, etc. Para tener acceso a ellos, solamente se tienen que pasar en los scripts como "*named inputs*". Pueden ser creados tanto desde la interfaz web como desde el SDK, como la mayoría de las operaciones del servicio.



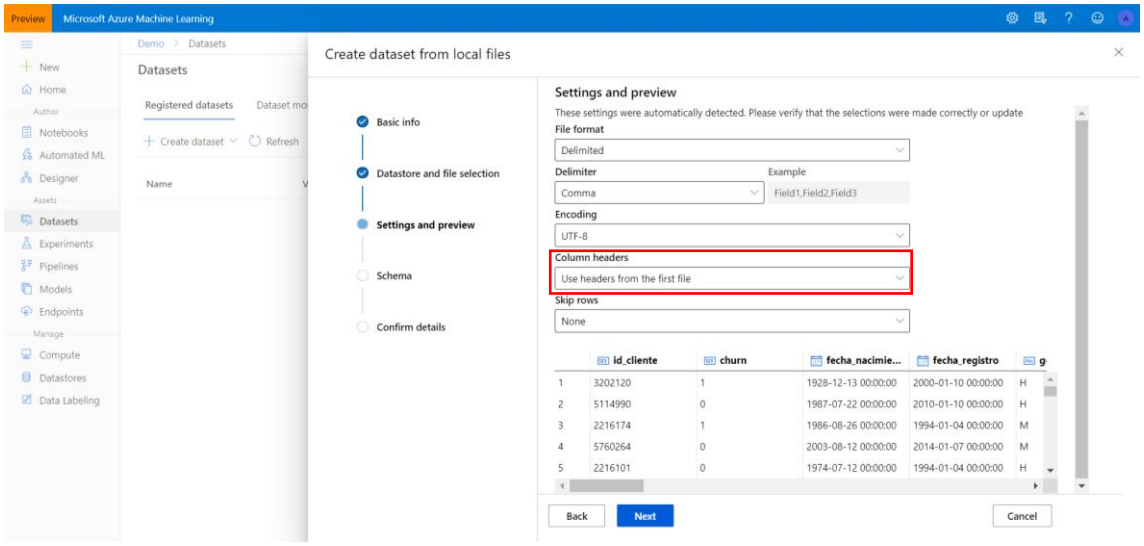
En este caso se crea un dataset (churn) desde un archivo local que además tiene formato tabular.



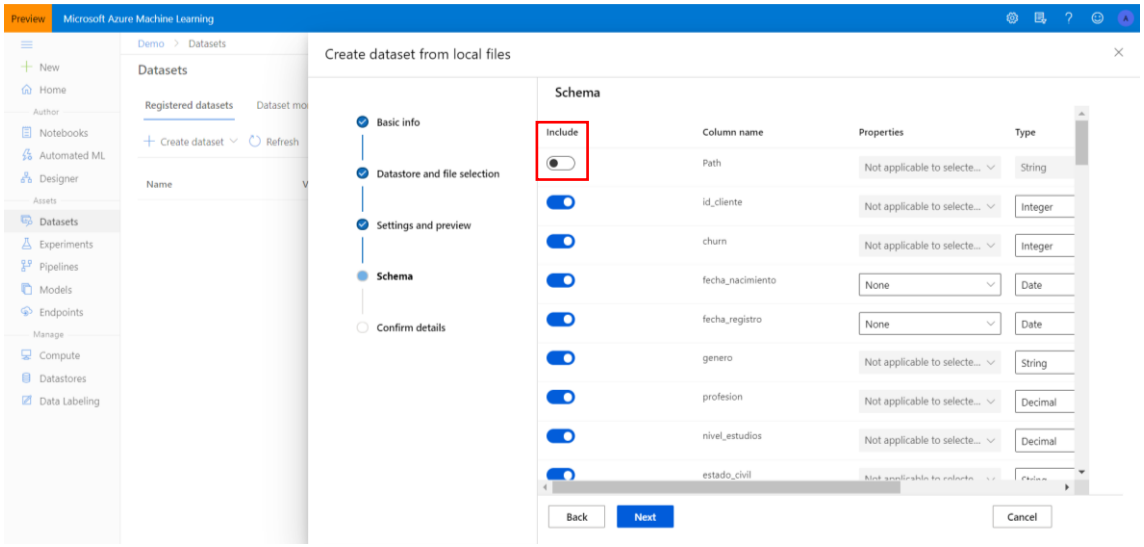
En la siguiente imagen aparece un datastore que se llama "churn", que ha sido creado durante la creación del dataset. Se han seguido los pasos indicados anteriormente.



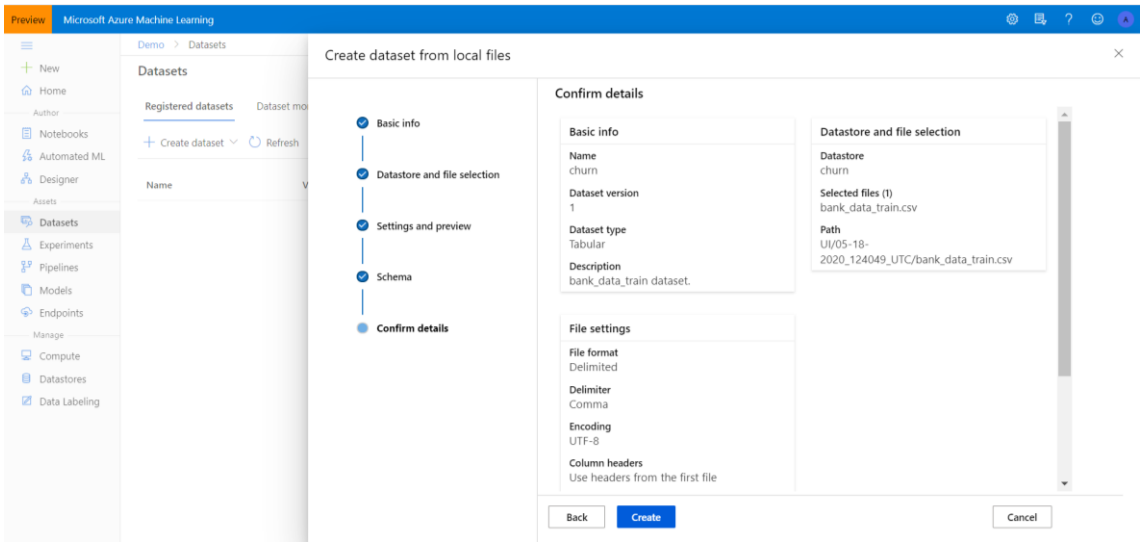
Se puede realizar una "Preview" de los datos, tal y como aparece en la siguiente imagen. Se especifica que se utilice la primera fila como nombre de las columnas.



En la imagen que aparece a continuación aparece un esquema de todas las variables que se van a incluir en el dataset. Se puede activar las columnas que se quieran importar al dataset, seleccionando el botón que aparece en la columna "Include".



Finalmente aparece un resumen con los detalles del dataset que se creará.



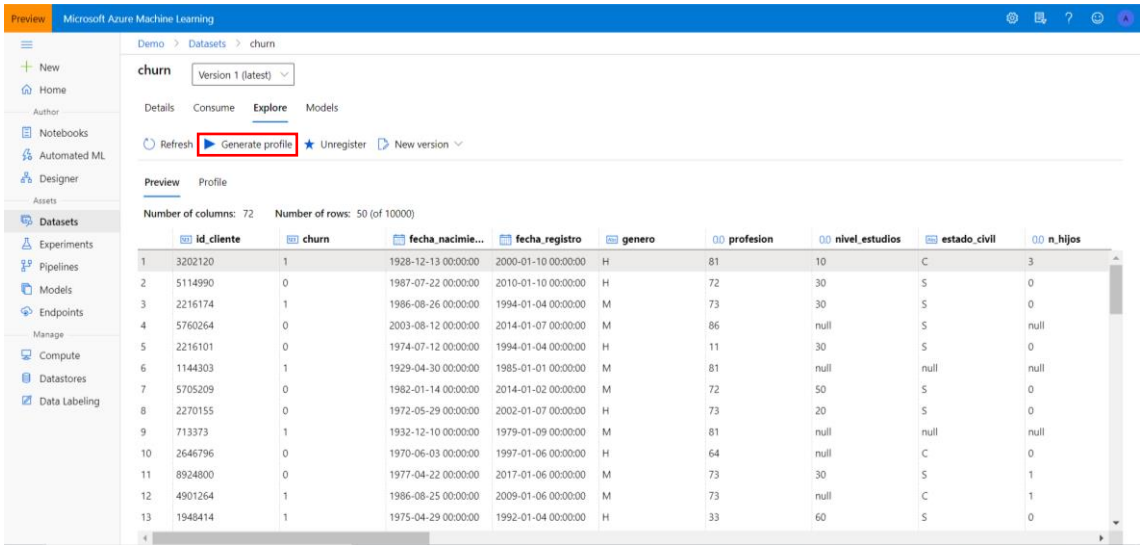
Existen dos tipos de datasets que se pueden registrar y utilizar:

1. **Datasets con una estructura tabular:** En este grupo entran todos los datasets que provengan de bases de datos relacionales. En este grupo los formatos más utilizados siguen siendo archivos separados por coma (CSV), Excel, etc.

2. **Datasets en ficheros que usan datos en formato semiestructurado o no estructurado,** como pueden ser ficheros JSON, XML, vídeo, etc.) Este tipo de formatos son cada vez más utilizados. En este caso, si se quieren entrenar modelos que usan estos formatos, se debe de registrar el dataset en formato **fichero**.

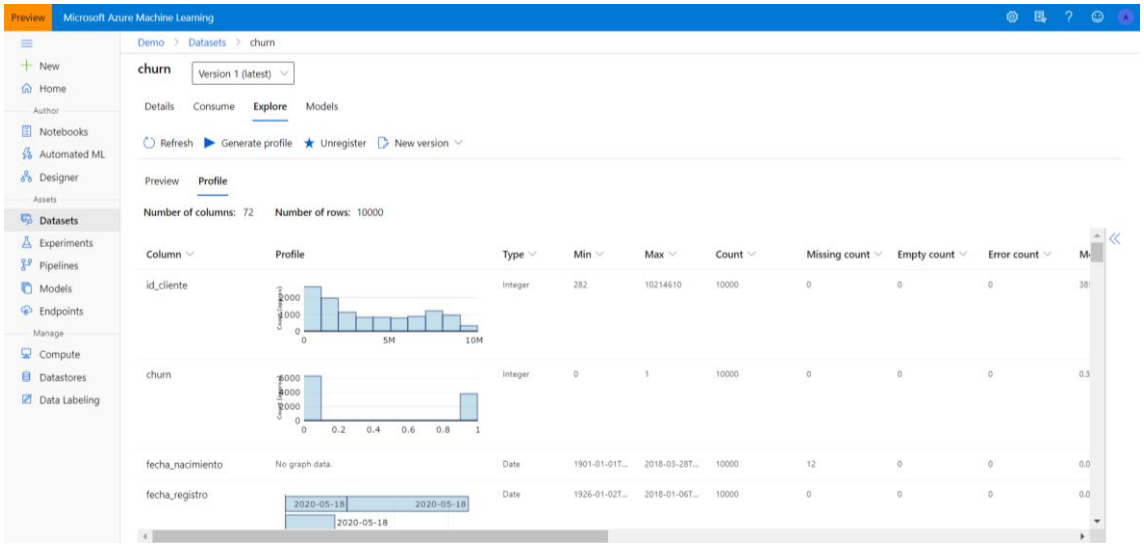
A continuación, una vez ya se tiene la configuración del tipo de dataset que se va a utilizar, se pasa a la siguiente fase donde se puede aplicar el perfilado de datos, realizar una exploración visual y estadística de los datos, etc. Esta fase no es más que la visualización de los datos que se usarán para construir el modelo. Es importante saber qué tipo de datos se tienen, la calidad de estos, etc.

Cabe destacar que en la siguiente imagen solo se muestran 10000 filas. Para tener disponibles todas las que contiene este dataset, se le tendrá que dar a "Generate profile". Hay dos pestañas, en la de "Preview" se puede realizar una previsualización de las diferentes columnas que hay en el dataset.



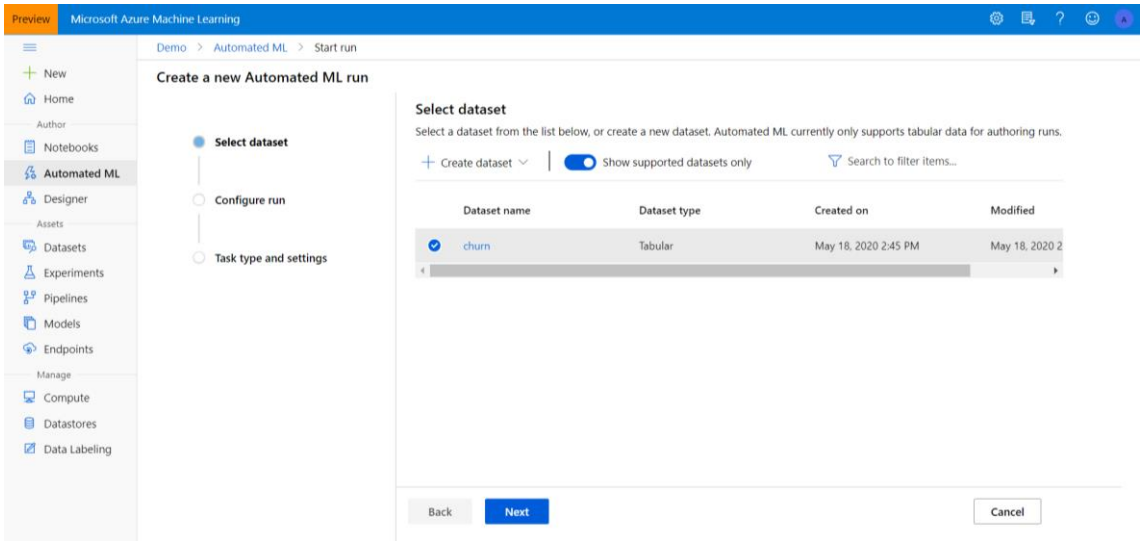
	id_cliente	churn	fecha_nacimie...	fecha_registro	genero	profesion	nivel_estudios	estado_civil	n_hijos
1	3202120	1	1928-12-13 00:00:00	2000-01-10 00:00:00	H	81	10	C	3
2	5114990	0	1987-07-22 00:00:00	2010-01-10 00:00:00	H	72	30	S	0
3	2216174	1	1986-08-26 00:00:00	1994-01-04 00:00:00	M	73	30	S	0
4	5760264	0	2003-08-12 00:00:00	2014-01-07 00:00:00	M	86	null	S	null
5	2216101	0	1974-07-12 00:00:00	1994-01-04 00:00:00	H	11	30	S	0
6	1144303	1	1929-04-30 00:00:00	1985-01-01 00:00:00	M	81	null	null	null
7	5705209	0	1982-01-14 00:00:00	2014-01-02 00:00:00	M	72	50	S	0
8	2270155	0	1972-05-29 00:00:00	2002-01-07 00:00:00	H	73	20	S	0
9	713373	1	1932-12-10 00:00:00	1979-01-09 00:00:00	M	81	null	null	null
10	2646796	0	1970-06-03 00:00:00	1997-01-06 00:00:00	H	64	null	C	0
11	8924800	0	1977-04-22 00:00:00	2017-01-06 00:00:00	M	73	30	S	1
12	4901264	1	1986-08-25 00:00:00	2009-01-06 00:00:00	M	73	null	C	1
13	1948414	1	1975-04-29 00:00:00	1992-01-04 00:00:00	H	33	60	S	0

En la pestaña de "Profile" aparecen diferentes estadísticas de cada columna.

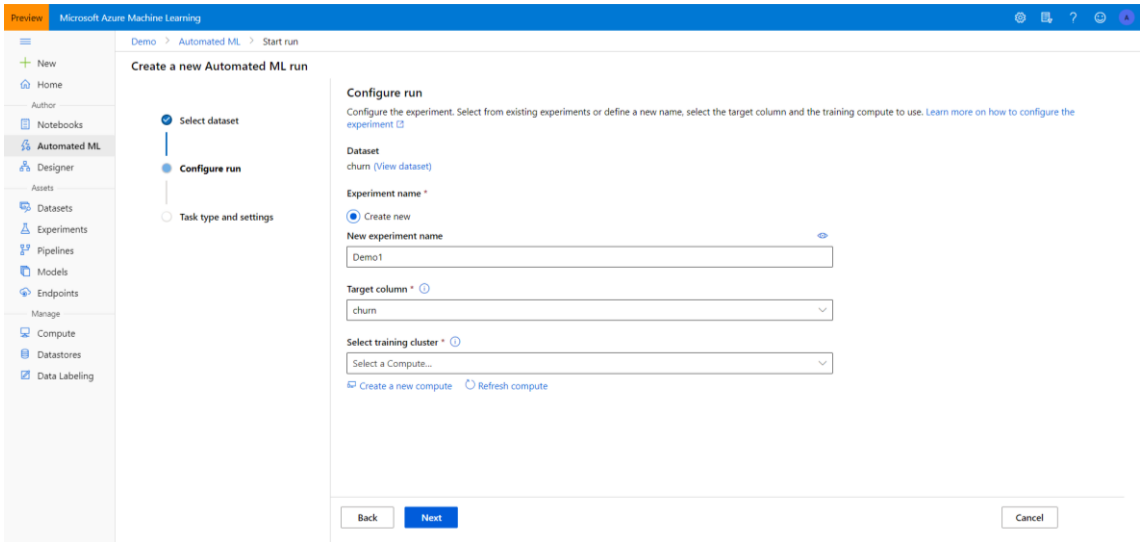


Como primer ejemplo, se va a realizar una demostración utilizando el método "**Automated ML".**

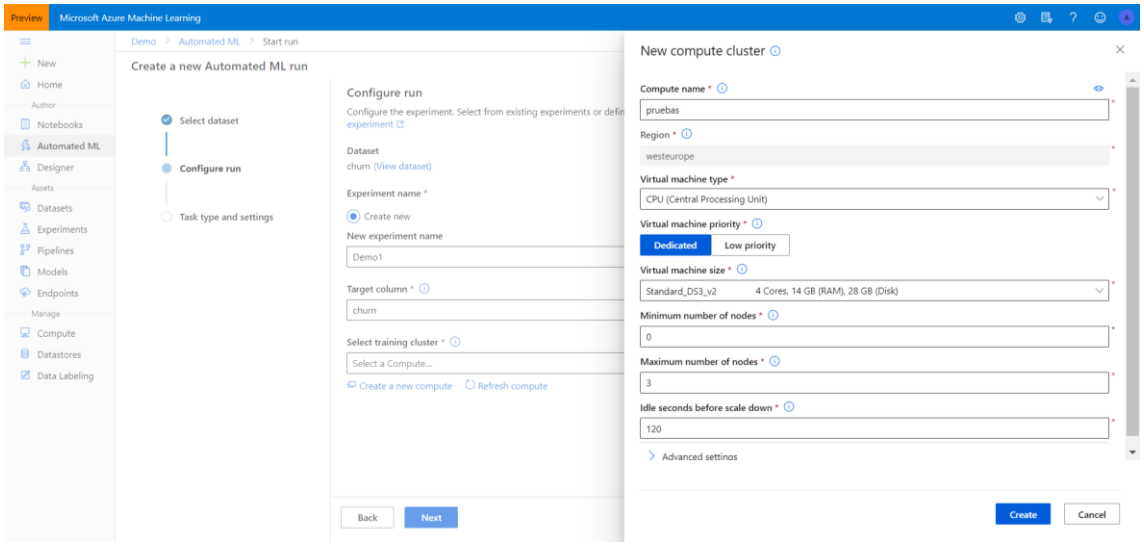
Se selecciona el dataset que se va a utilizar. En este caso, se usará el dataset de "churn", el cual contiene diferente información de los clientes de un banco.



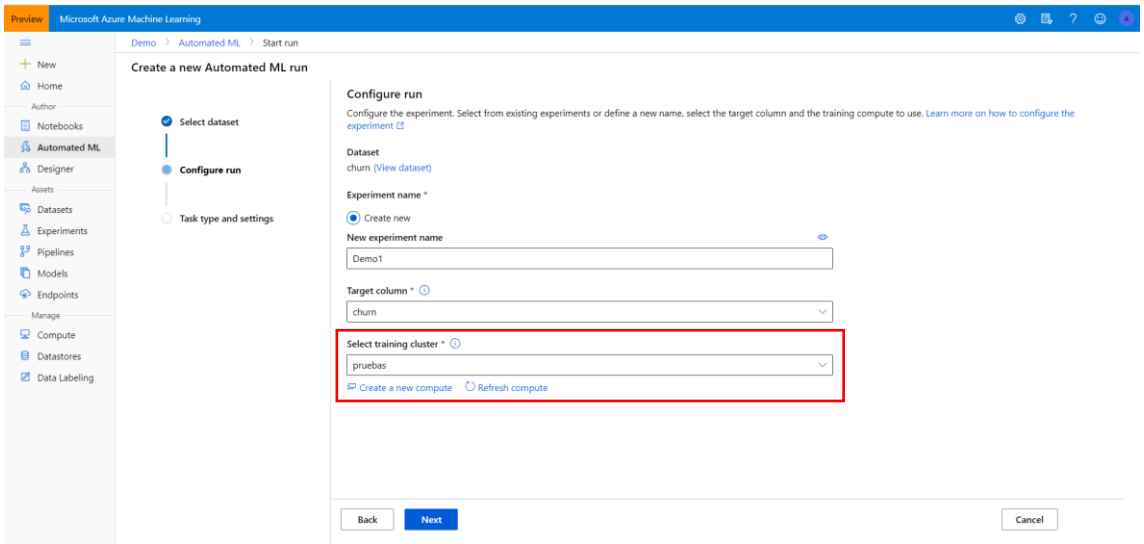
Después se crea un nuevo "Experimento", al que se ha llamado "Demo1". La columna target que será la que se quiera predecir es la de "churn". También se tiene que seleccionar un "Compute target".



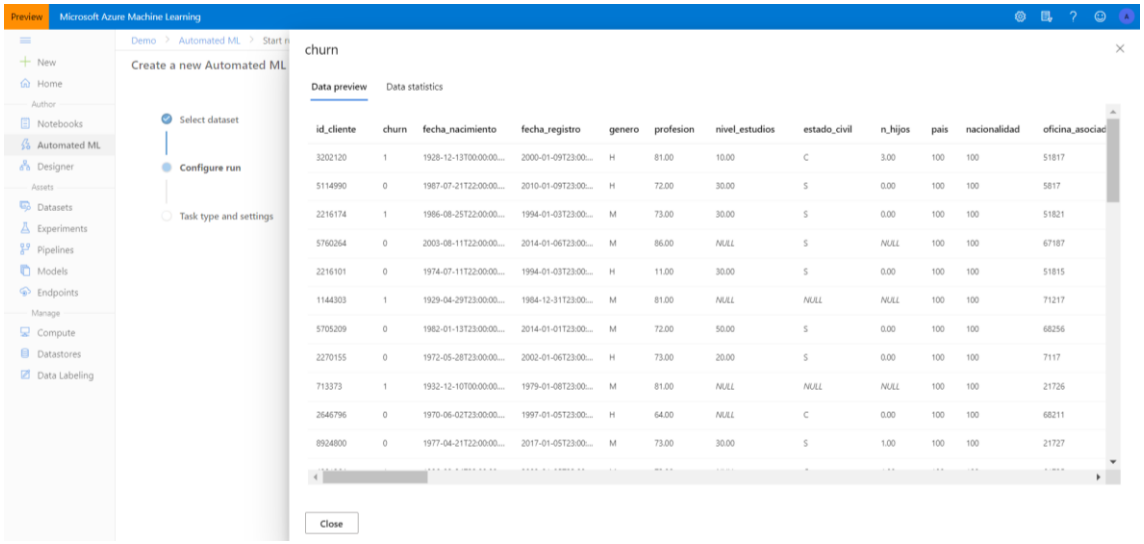
Por ello, se configura un nuevo "Compute cluster" con las siguientes características:

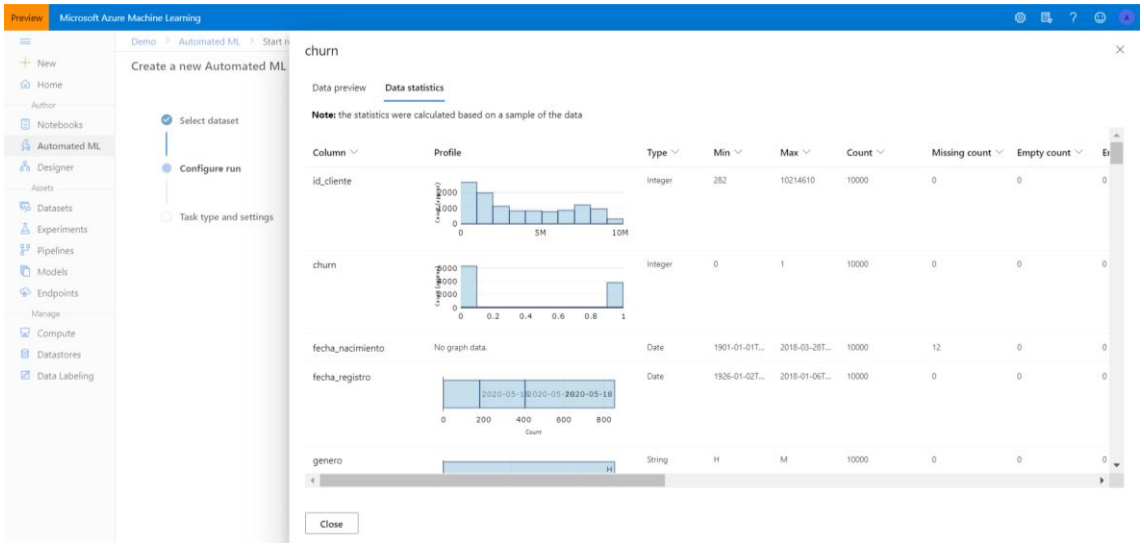


Después se selecciona el nuevo "Compute cluster" creado en el paso anterior.

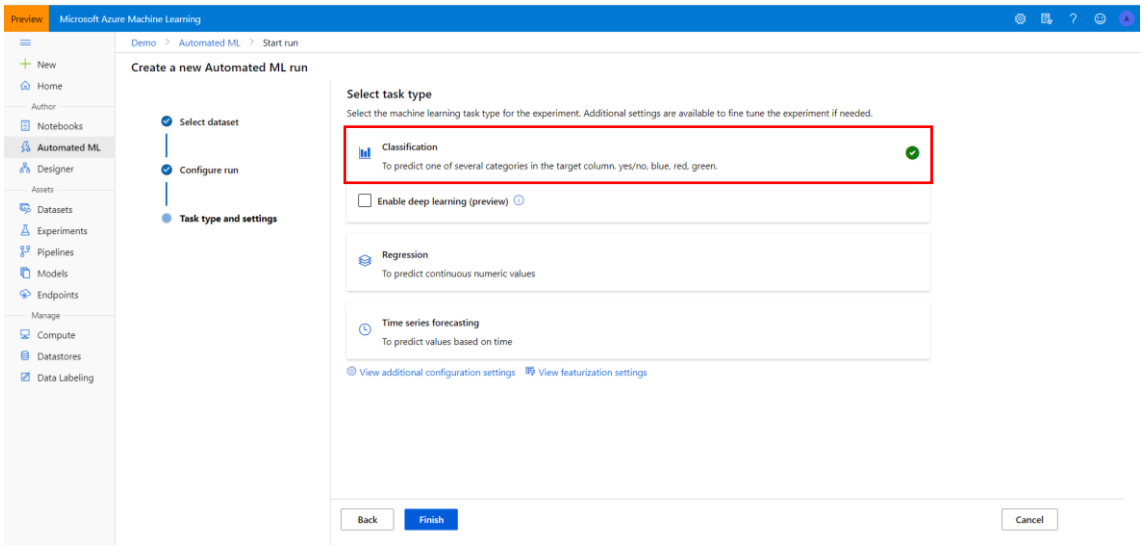


Como al crear el dataset, se puede hacer una "Preview" de los datos con los que vamos a trabajar, así como sacar algunas estadísticas de los mismos.

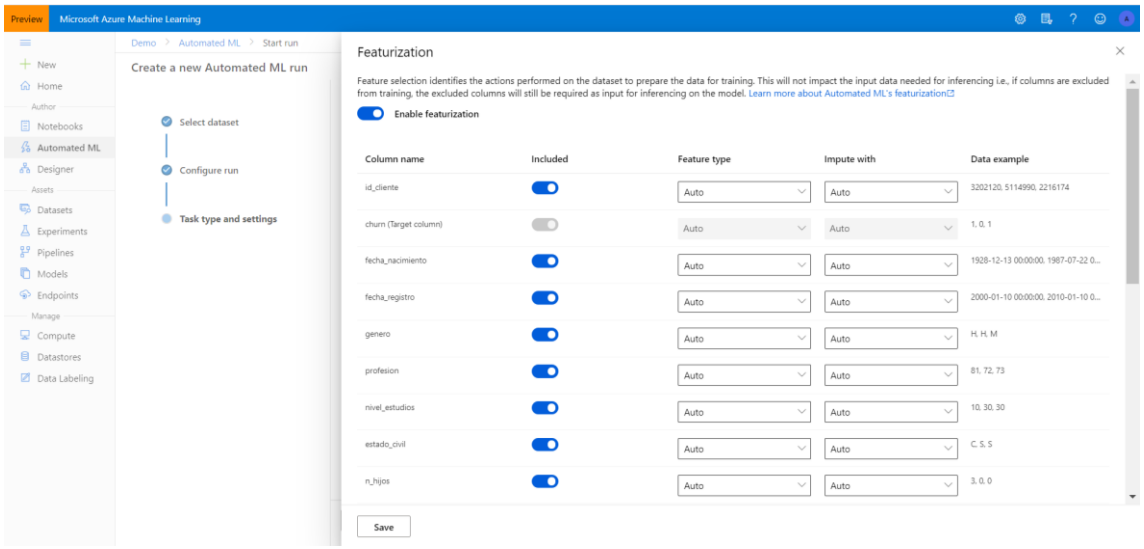




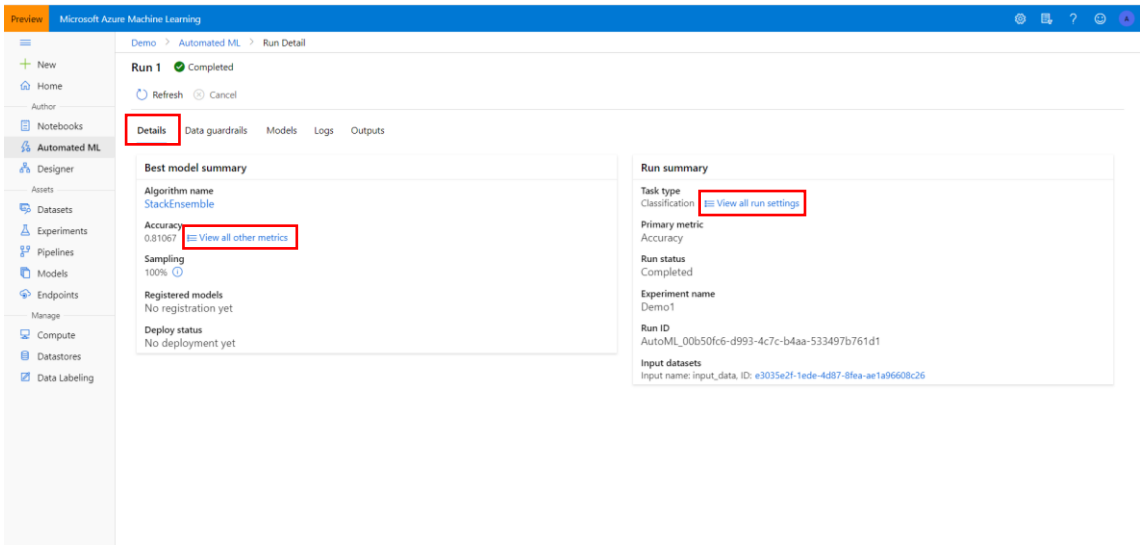
Se selecciona el tipo de Modelo de Machine Learning, en este caso es un modelo de Clasificación. Se quiere predecir si un cliente abandonará el banco o no.



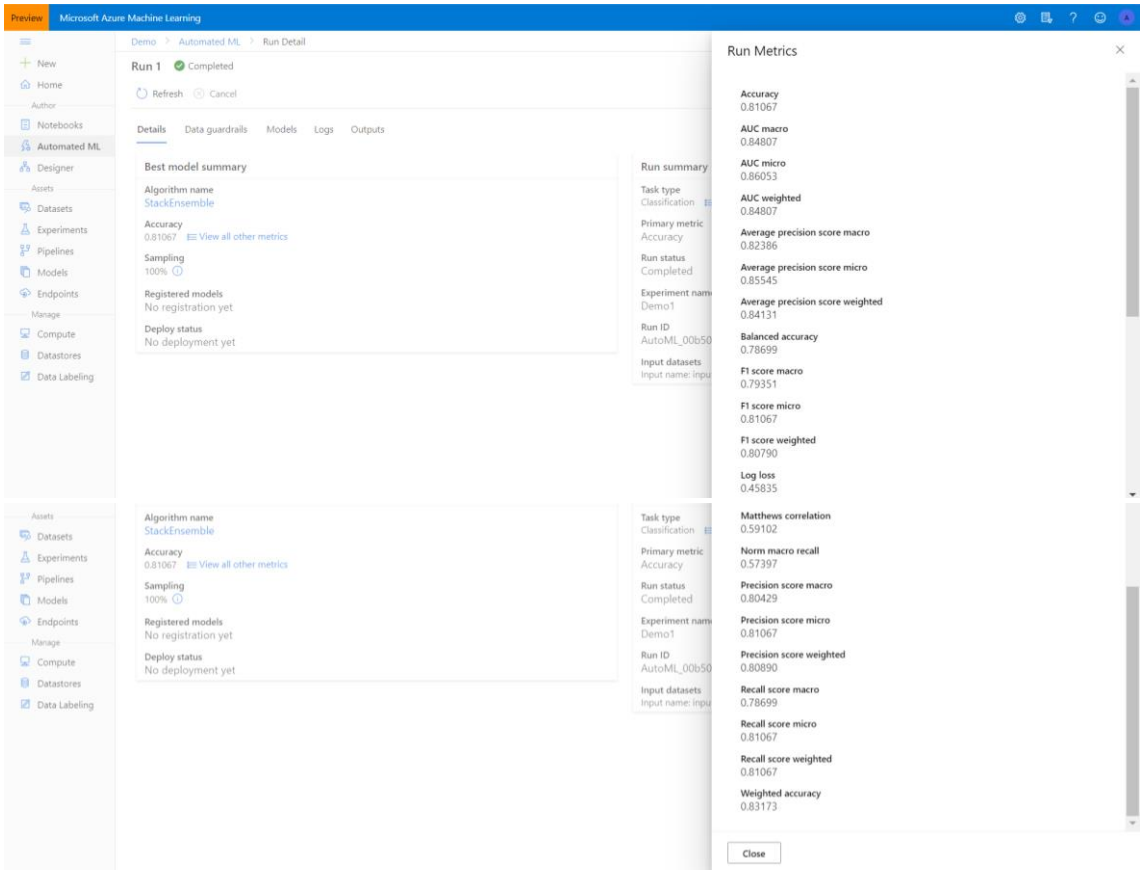
En cuanto al resto de "Settings" se deja todo tal y como sale por defecto.



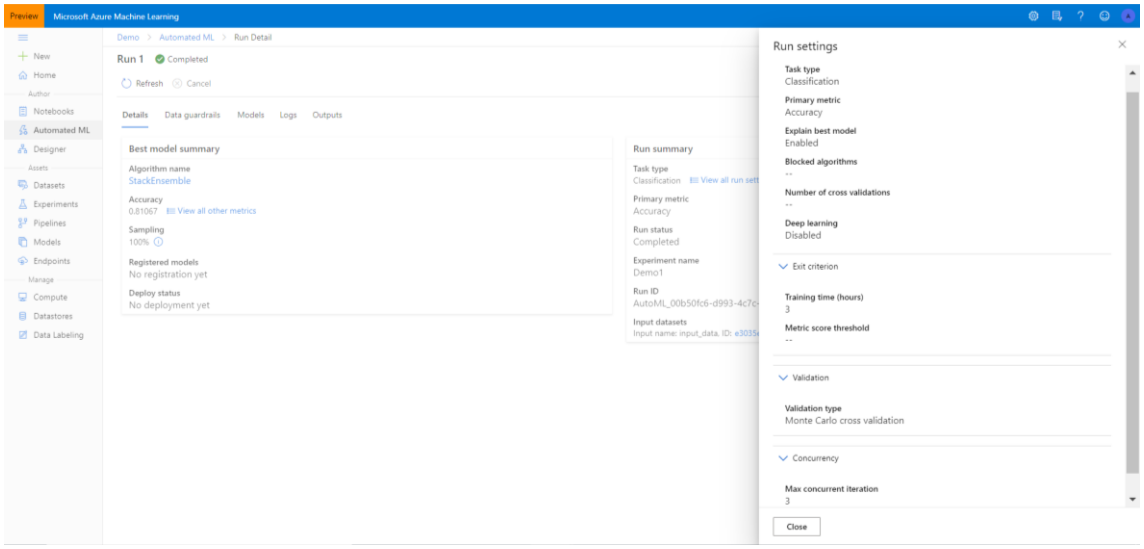
Una vez que el proceso de Automated ML ha terminado, en la pestaña de detalles del proceso (Details), se puede ver un resumen, tanto del mejor modelo resultante, como de los detalles del propio proceso que se ha configurado. En este caso, el mejor modelo ha sido "Stack Ensemble", con una precisión o accuracy de 0.81067.



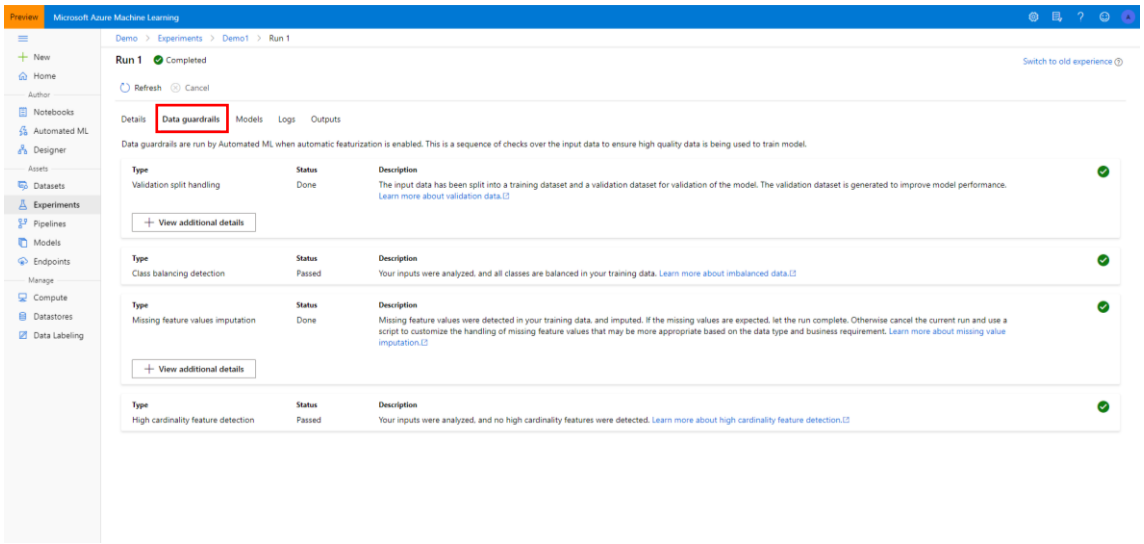
Por defecto aparece como métrica de evaluación del mejor modelo, el nivel de precisión o accuracy. La precisión es el porcentaje de las etiquetas de predicción que coinciden exactamente con las etiquetas verdaderas. También es posible observar otras métricas, pulsando en "View all other metrics". Puede consultar el siguiente link para más información sobre las diferentes métricas que se computan: https://docs.microsoft.com/es-es/azure/machine-learning/how-to-understand-automated-ml#view-the-run



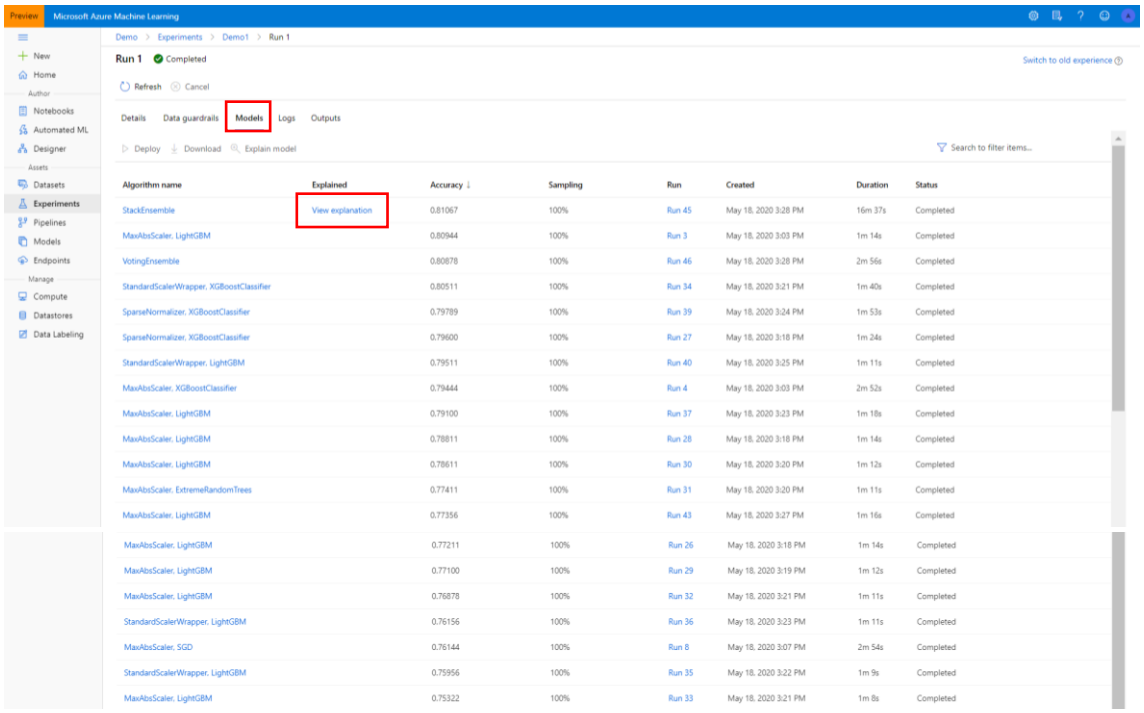
Además, pulsando en "View all run settings" también se pueden ver los detalles de cómo ha sido configurado el proceso.

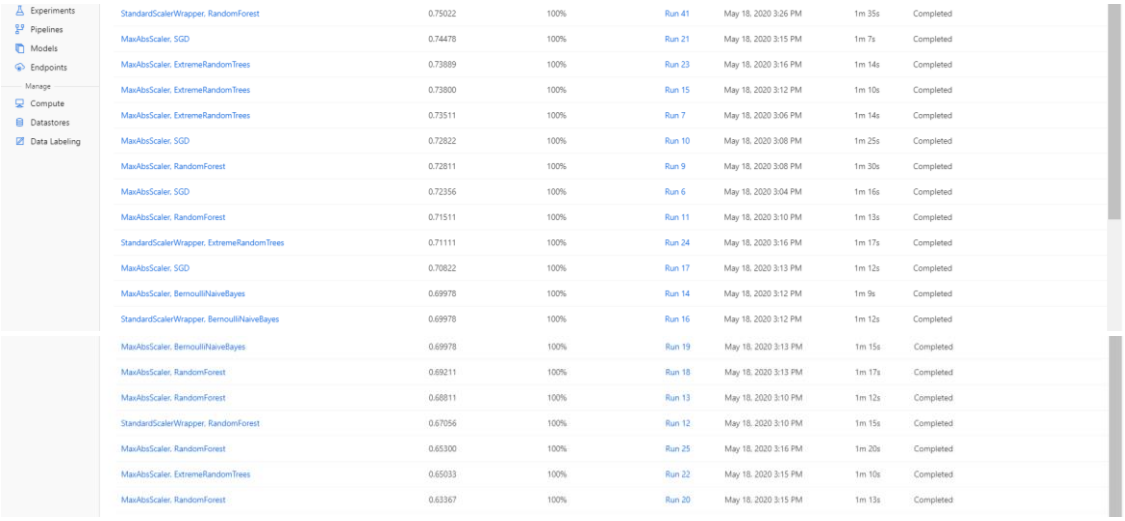


En la siguiente pestaña (Data guardrails), aparece una secuencia de verificaciones sobre los datos de entrada para garantizar que se utilicen datos de alta calidad para entrenar el modelo.

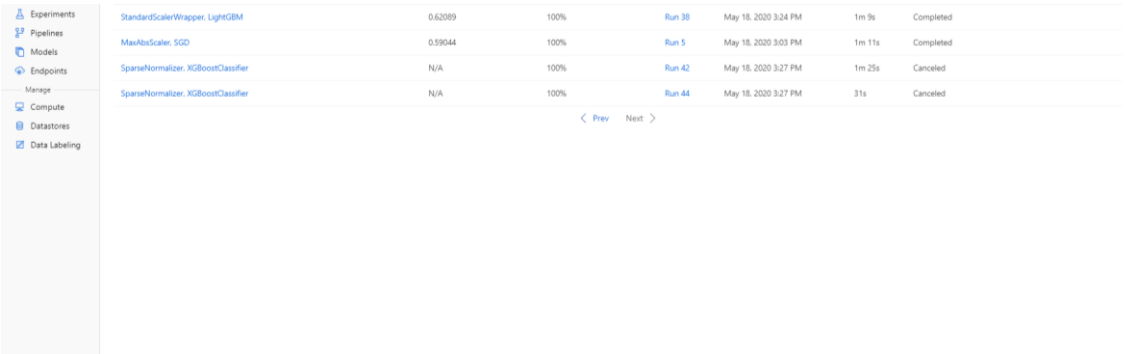


A continuación, en la pestaña "Models" aparecen los resultados obtenidos.



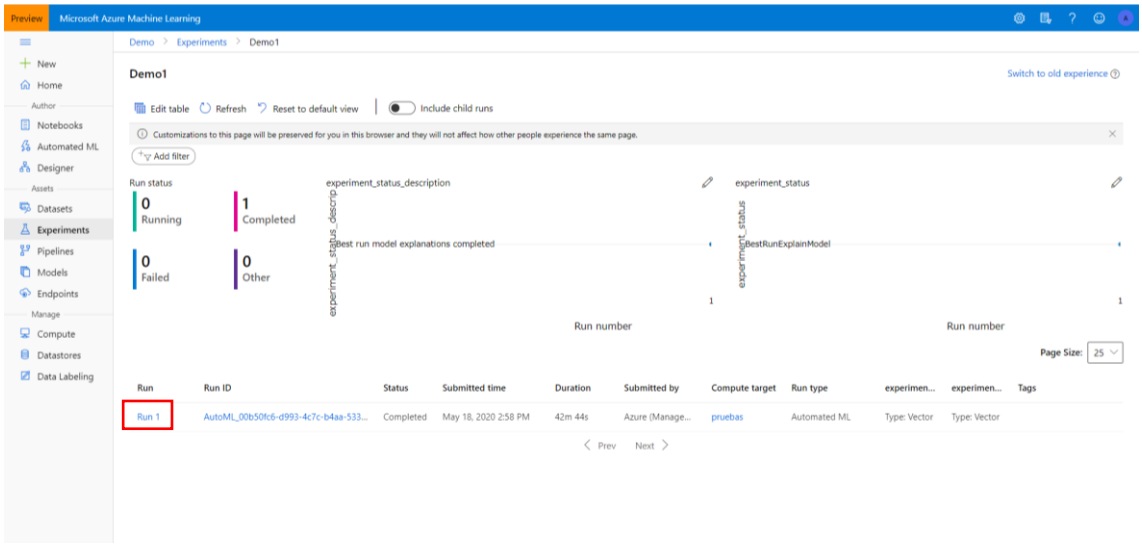


StandardScalerWrapper, RandomForest	0.75022	100%	Run 41	May 18, 2020 3:26 PM	1m 35s	Completed
MaxAbsScaler, SGD	0.74478	100%	Run 21	May 18, 2020 3:15 PM	1m 7s	Completed
MaxAbsScaler, ExtremeRandomTrees	0.73889	100%	Run 23	May 18, 2020 3:16 PM	1m 14s	Completed
MaxAbsScaler, ExtremeRandomTrees	0.73800	100%	Run 15	May 18, 2020 3:12 PM	1m 10s	Completed
MaxAbsScaler, ExtremeRandomTrees	0.73511	100%	Run 7	May 18, 2020 3:06 PM	1m 14s	Completed
MaxAbsScaler, SGD	0.72822	100%	Run 10	May 18, 2020 3:08 PM	1m 25s	Completed
MaxAbsScaler, RandomForest	0.72811	100%	Run 9	May 18, 2020 3:08 PM	1m 30s	Completed
MaxAbsScaler, SGD	0.72356	100%	Run 6	May 18, 2020 3:04 PM	1m 16s	Completed
MaxAbsScaler, RandomForest	0.71511	100%	Run 11	May 18, 2020 3:10 PM	1m 13s	Completed
StandardScalerWrapper, ExtremeRandomTrees	0.71111	100%	Run 24	May 18, 2020 3:16 PM	1m 17s	Completed
MaxAbsScaler, SGD	0.70822	100%	Run 17	May 18, 2020 3:13 PM	1m 12s	Completed
MaxAbsScaler, BernoulliNaiveBayes	0.69978	100%	Run 14	May 18, 2020 3:12 PM	1m 9s	Completed
StandardScalerWrapper, BernoulliNaiveBayes	0.69978	100%	Run 16	May 18, 2020 3:12 PM	1m 12s	Completed
MaxAbsScaler, BernoulliNaiveBayes	0.69978	100%	Run 19	May 18, 2020 3:13 PM	1m 15s	Completed
MaxAbsScaler, RandomForest	0.69211	100%	Run 18	May 18, 2020 3:13 PM	1m 17s	Completed
MaxAbsScaler, RandomForest	0.68811	100%	Run 13	May 18, 2020 3:10 PM	1m 12s	Completed
StandardScalerWrapper, RandomForest	0.67056	100%	Run 12	May 18, 2020 3:10 PM	1m 15s	Completed
MaxAbsScaler, RandomForest	0.65300	100%	Run 25	May 18, 2020 3:16 PM	1m 20s	Completed
MaxAbsScaler, ExtremeRandomTrees	0.65033	100%	Run 22	May 18, 2020 3:15 PM	1m 10s	Completed
MaxAbsScaler, RandomForest	0.63367	100%	Run 20	May 18, 2020 3:15 PM	1m 13s	Completed

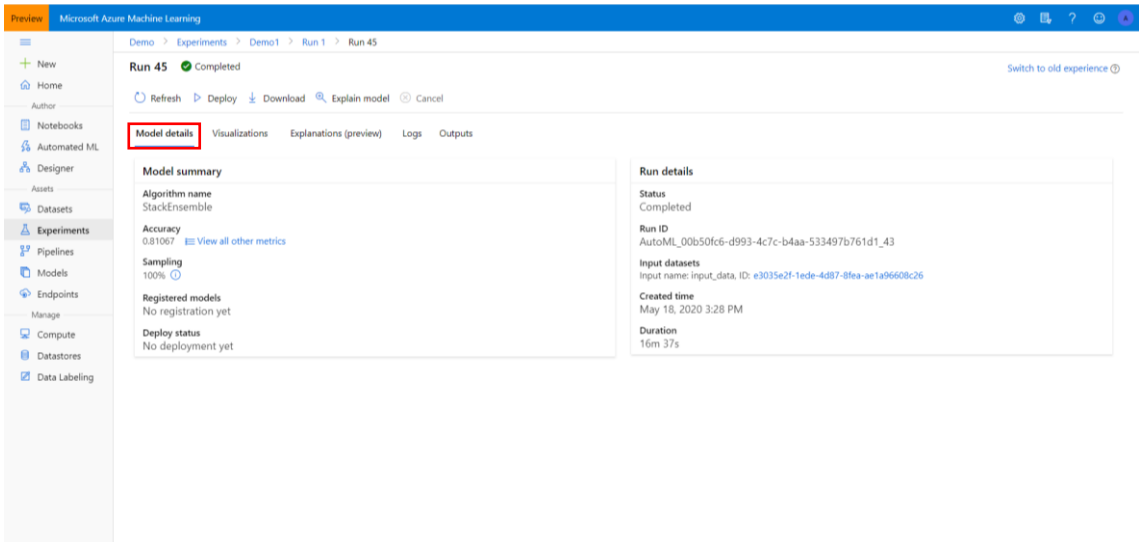


StandardScalerWrapper, LightGBM	0.62089	100%	Run 38	May 18, 2020 3:24 PM	1m 9s	Completed
MaxAbsScaler, SGD	0.59044	100%	Run 5	May 18, 2020 3:03 PM	1m 11s	Completed
SparseNormalizer, XGBoostClassifier	N/A	100%	Run 42	May 18, 2020 3:27 PM	1m 25s	Canceled
SparseNormalizer, XGBoostClassifier	N/A	100%	Run 44	May 18, 2020 3:27 PM	31s	Canceled

Azure ML te permite realizar un ánalisis más profundo del mejor modelo resultante, en este caso "Stack Ensemble", por lo que después de ejecutar un experimento de Automate Machine Learning, puede encontrar un historial de las ejecuciones en el área de trabajo de Machine Learning en el que esté trabajando.



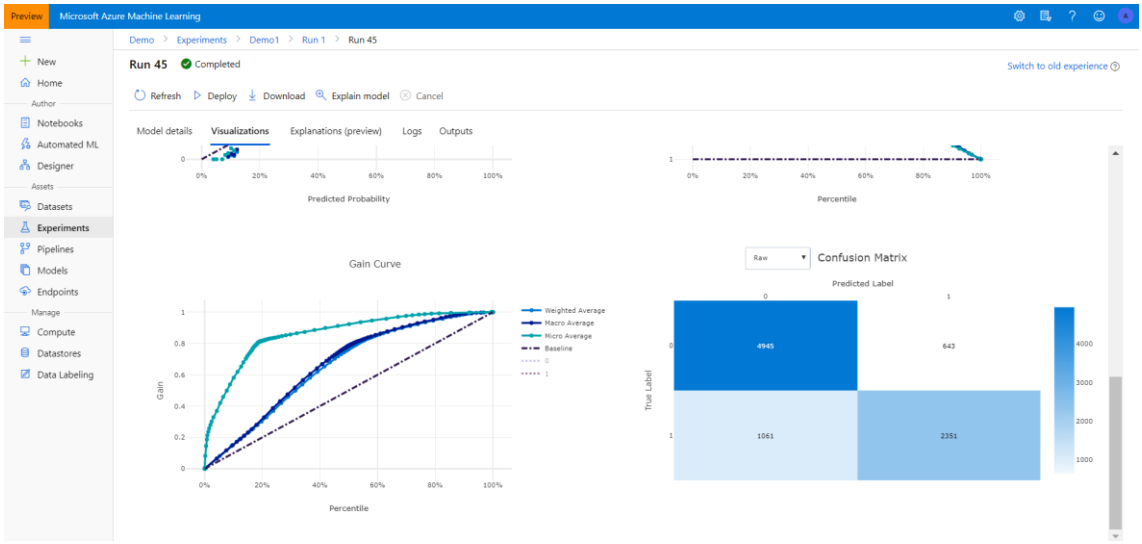
En la pestaña de "Model details" aparece la configuración del modelo final.



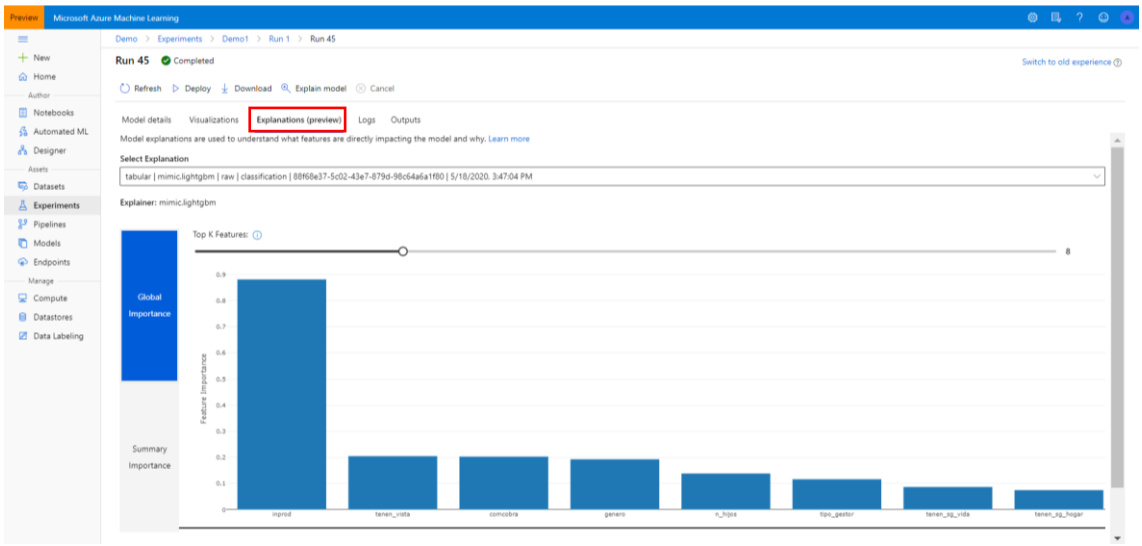
En la siguiente pestaña de"Visualizations", aparecen diferentes gráficos de algunas métricas tales como la Curva ROC, la Matriz de Confusión, etc. Con estas visualizaciones se busca una mejor comprensión del rendimiento del modelo mediante gráficos.

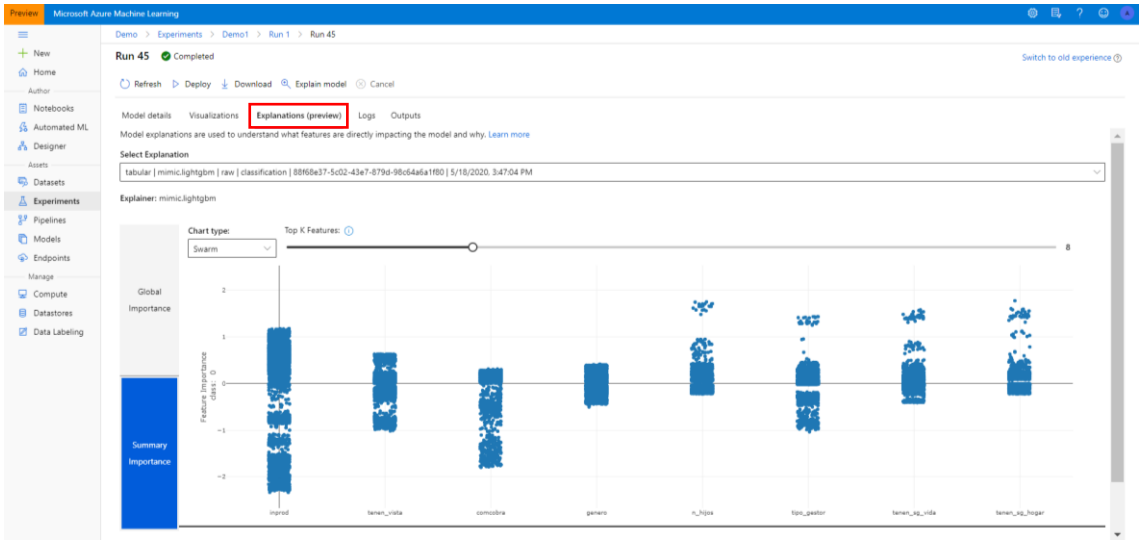

Preview
Microsoft Azure Machine Learning
Demo > Experiments > Demo1 > Run 1 > Run 45
Run 45
Completed
Switch to old experience
Refresh
Deploy
Download
Explain model
Cancel
Model details
Visualizations
Explanations (preview)
Logs
Outputs
Automated ML provides charts for better understanding of model performance. Learn more on how to analyze your model
Precision-Recall
ROC
Calibration Curve
Lift Curve
Recall
Precision
False Positive Rate
True Positive Rate
Weighted Average
Macro Average
Micro Average
Ideal
Random classifier


Preview
Microsoft Azure Machine Learning
Demo > Experiments > Demo1 > Run 1 > Run 45
Run 45
Completed
Switch to old experience
Refresh
Deploy
Download
Explain model
Cancel
Model details
Visualizations
Explanations (preview)
Logs
Outputs
Recall
False Positive Rate
Calibration Curve
Lift Curve
Actual Probability
Predicted Probability
Lift
Percentile
Weighted Average
Macro Average
Micro Average
Ideal
Baseline
Gain Curve
Raw
Confusion Matrix



En la pestaña de "Explanation (preview)" del modelo que se utilizan para comprender qué características están afectando directamente al modelo, qué variables tienen más importancia a la hora de predecir el "churn rate", etc.

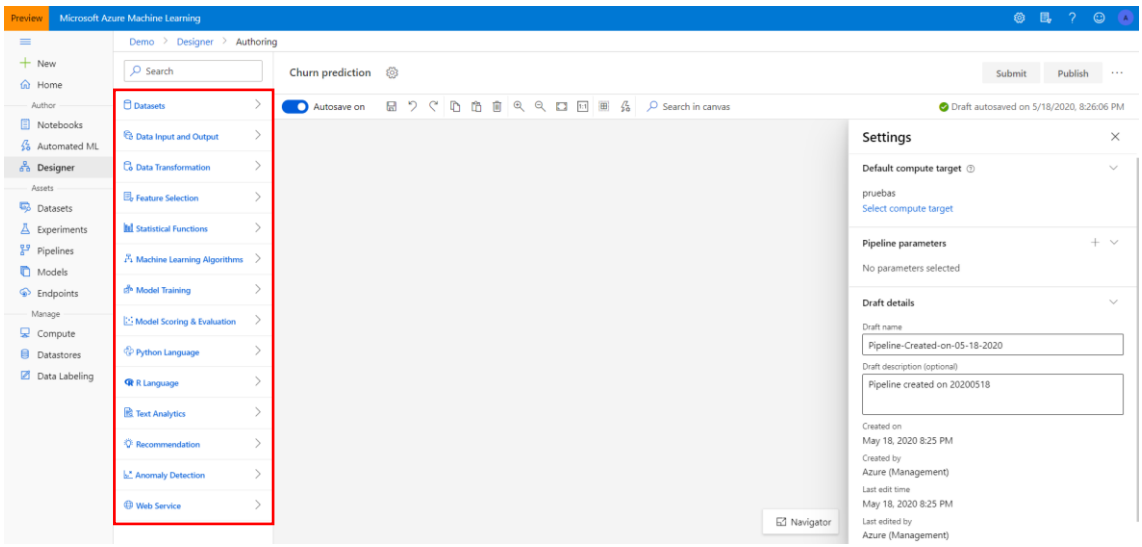



Preview
Microsoft Azure Machine Learning
Demo > Experiments > Demo1 > Run 1 > Run 45
New
Home
Author
Notebooks
Automated ML
Designer
Assets
Datasets
Experiments
Pipelines
Models
Endpoints
Manage
Compute
Datastores
Data Labeling
Run 45
Completed
Switch to old experience
Refresh
Deploy
Download
Explain model
Cancel
Model details
Visualizations
Explanations (preview)
Logs
Outputs
Model explanations are used to understand what features are directly impacting the model and why. Learn more
Select Explanation
tabular | mimic.lightgbm | raw | classification | 88f68e37-5c02-43e7-879d-98c64a6a1f80 | 5/18/2020, 3:47:04 PM
Explainer: mimic.lightgbm
Chart type:
Swarm
Top K Features:
8
Global Importance
Summary Importance
Feature Importance
class: 0
2
1
0
−1
−2
inprod
tenen_vivta
comcobra
genero
n_hijos
tipo_gestor
tenen_sg_vida
tenen_sg_hogar

7. EJEMPLO 2

Para esta segunda demostración se va a utilizar el mismo dataset de "churn", pero en este ejemplo se usará el método "Designer" mediante el diseñador de Azure ML (versión preliminar).

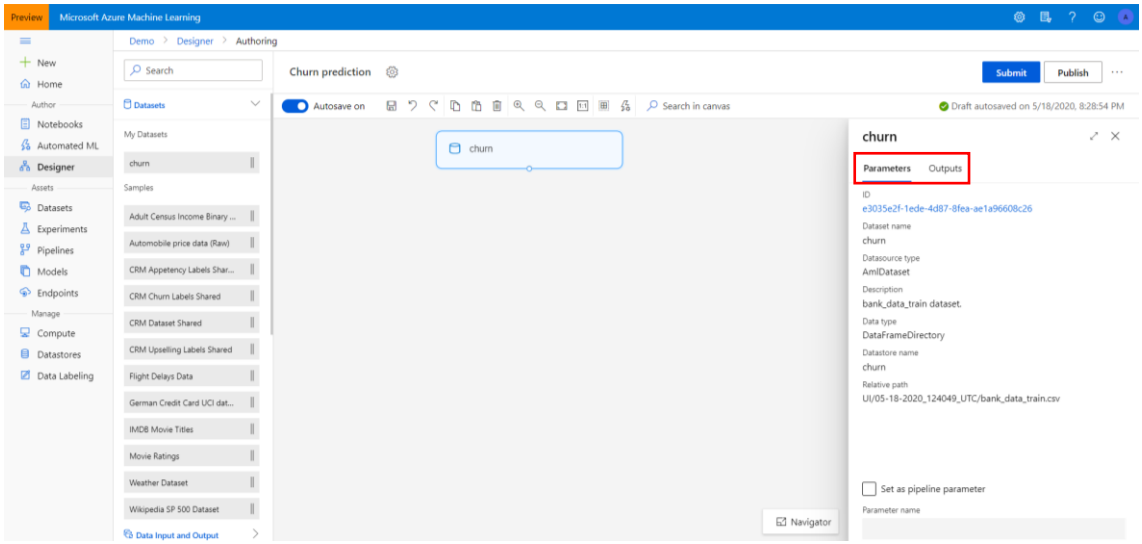
Antes de nada se crea un nuevo Pipeline, para ello se selecciona la opción "Easy-to-use prebuilt modules". Después la interfaz que nos sale es la siguiente:



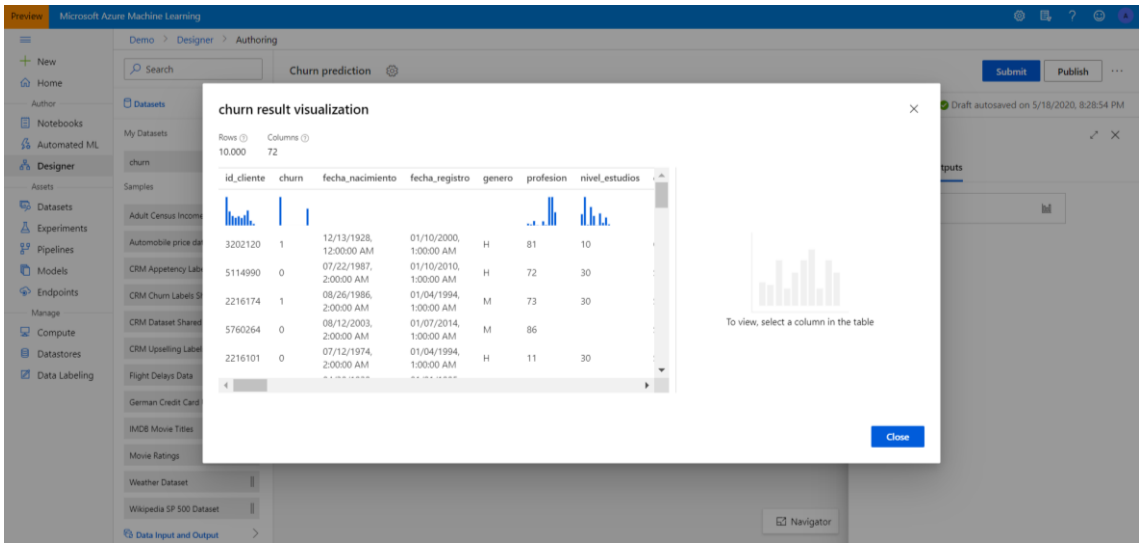
En ella se pueden observar distintas opciones para crear el Workflow del Pipeline del modelo que se quiera crear, en este caso un modelo de Clasificación. Entre ellas destacan las opciones de Preprocesado de datos, tales como transformación de variables, eliminación de valores nulos, estandarización de variables, selección de variables relevantes, etc.

En primer lugar se pone como valor de entrada el dataset de "churn", que será sobre el que se trabaje. En la siguiente imagen se pueden observar dos pestañas diferentes, una de "Parameters" y otra de "Outputs".

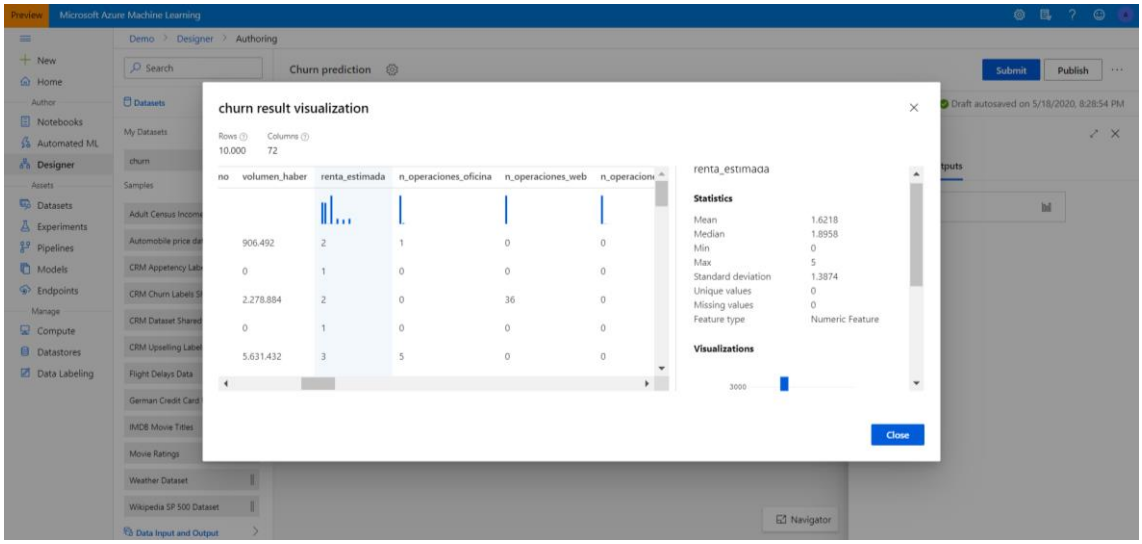
En la pestaña de parámetros se encuentran diferentes características relativas al dataset de "churn".

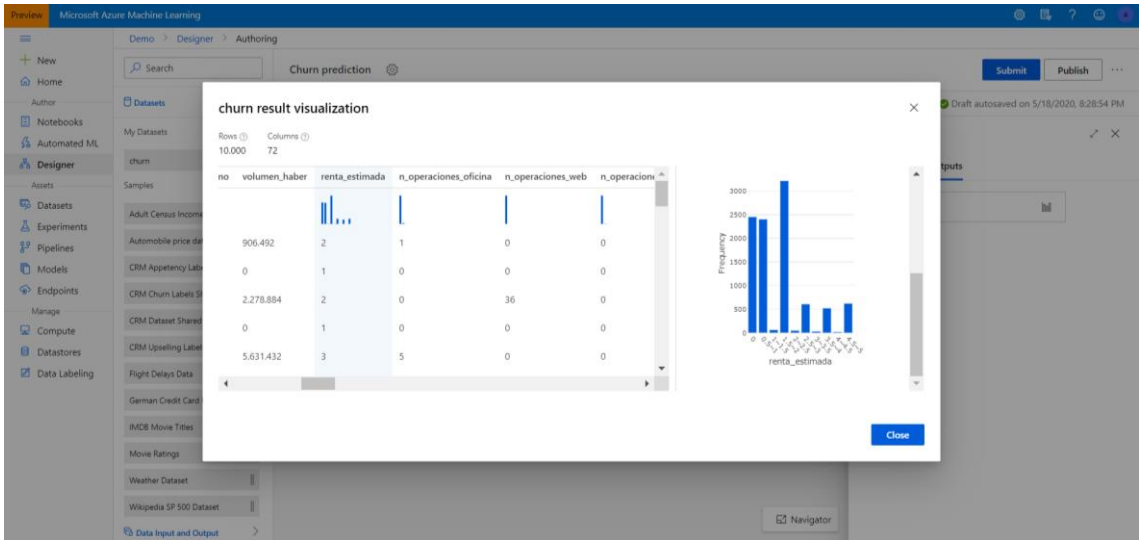


En la parte de "Outputs" se puede realizar una visualización de las diferentes variables existentes en el dataset elegido.

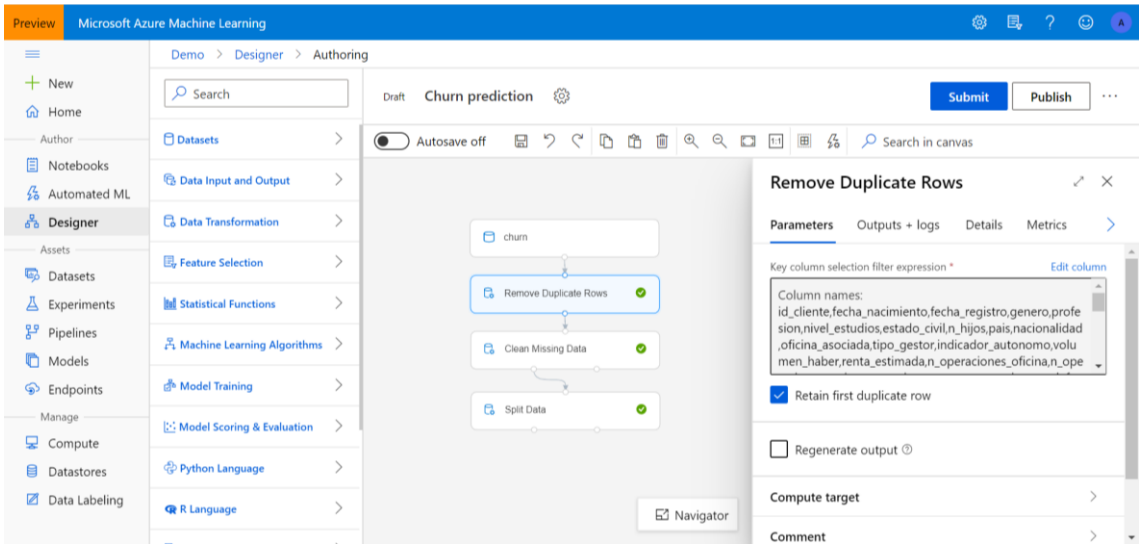


También se puede realizar una previsualización de las principales estadísticas y distribución de la columna que se quiera. En este ejemplo se quiere ver con más detalle la variable "renta_estimada".

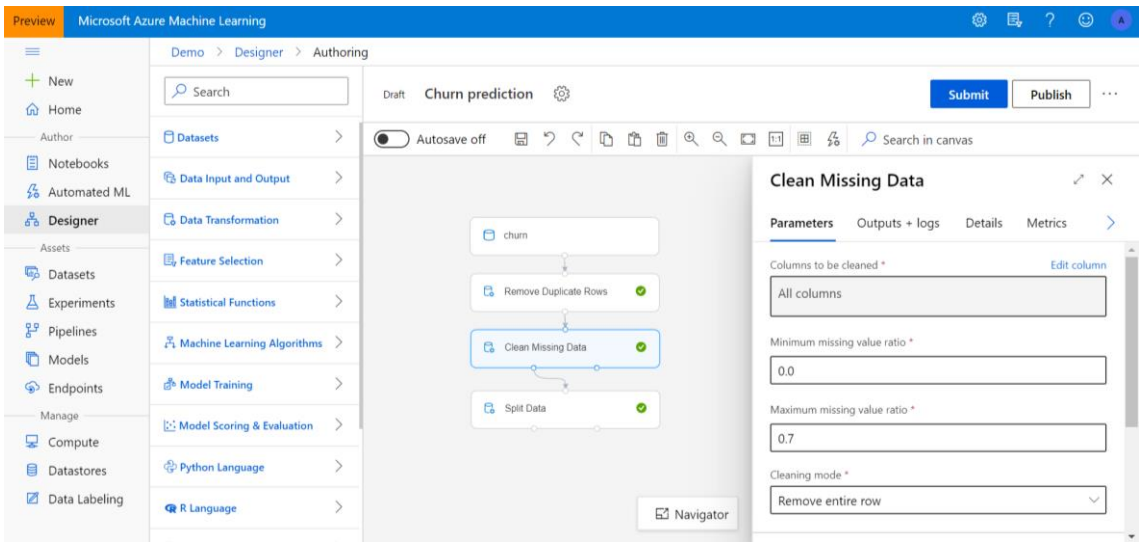




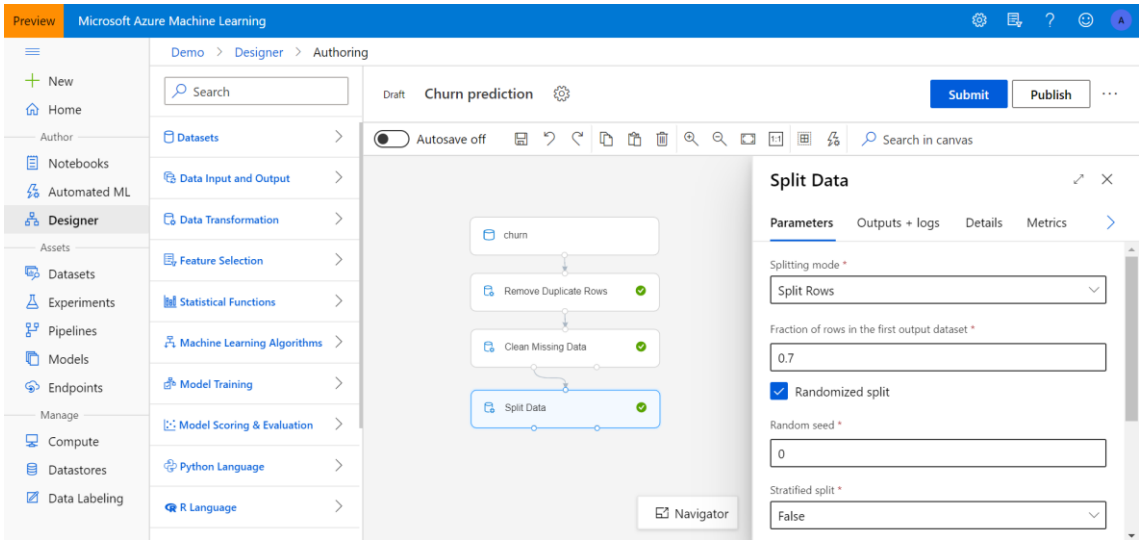
Luego se procede a eliminar los valores de aquellas filas que estén duplicadas. Se seleccionan todas las variables menos la variable de respuesta “churn”.



Después se pasa a la fase de la limpieza de missing values. Se eliminan todas las filas que tengan un porcentaje de missing values del 70 % o más.

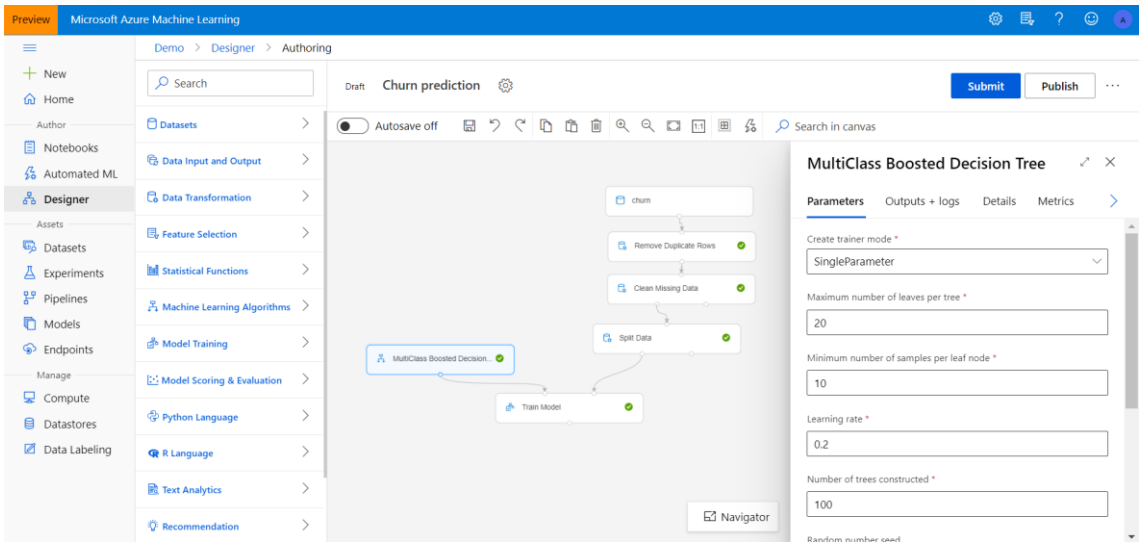


A continuación se divide el dataset en 70 % train y 30 % test. Esto se hace para utilizar la parte de train para el entrenamiento del modelo, y la parte de test para evaluar cómo de bueno es el modelo que se ha generado.

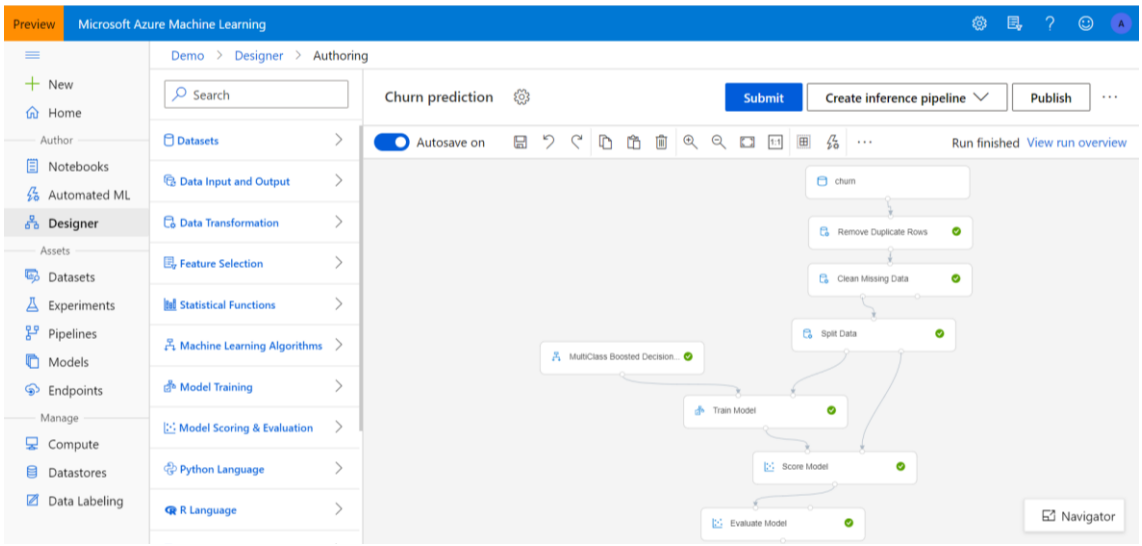


Para este ejemplo se ha elegido el algoritmo de "Multiclass Boosted Decision Tree". Se usa este algoritmo para crear un ensamblamiento de árboles de decisión utilizando el método "Boosting". "Boosting" significa que cada árbol depende del árbol que se haya construido anteriormente. Es un método de aprendizaje de conjuntos en que el segundo árbol corrige los errores del primer árbol, el tercer árbol corrige los errores del primer y del segundo árbol y así sucesivamente. Las predicciones se basan en el conjunto de árboles juntos.

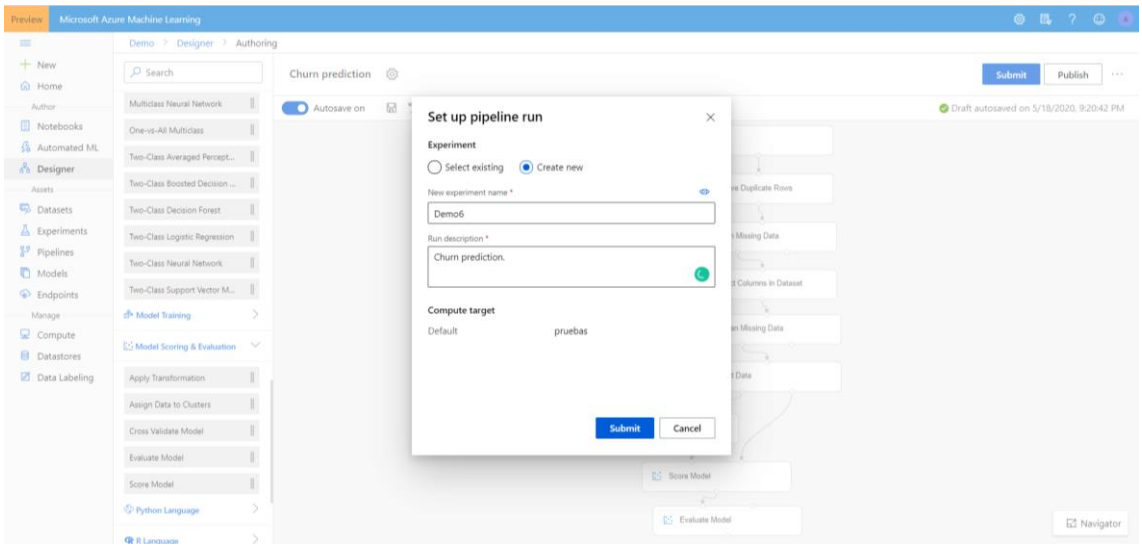
Este método forma parte de los algoritmos de "Supervised Machine Learning", por lo que requiere de una variable de respuesta definida, que además debe de ser numérica, en este caso "churn".



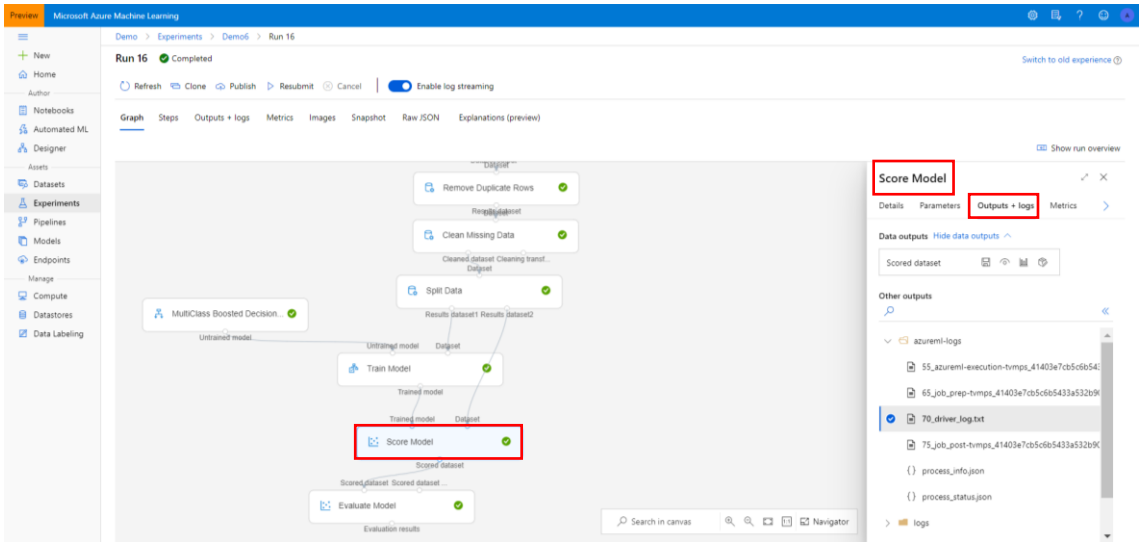
El workflow final quedaría así:

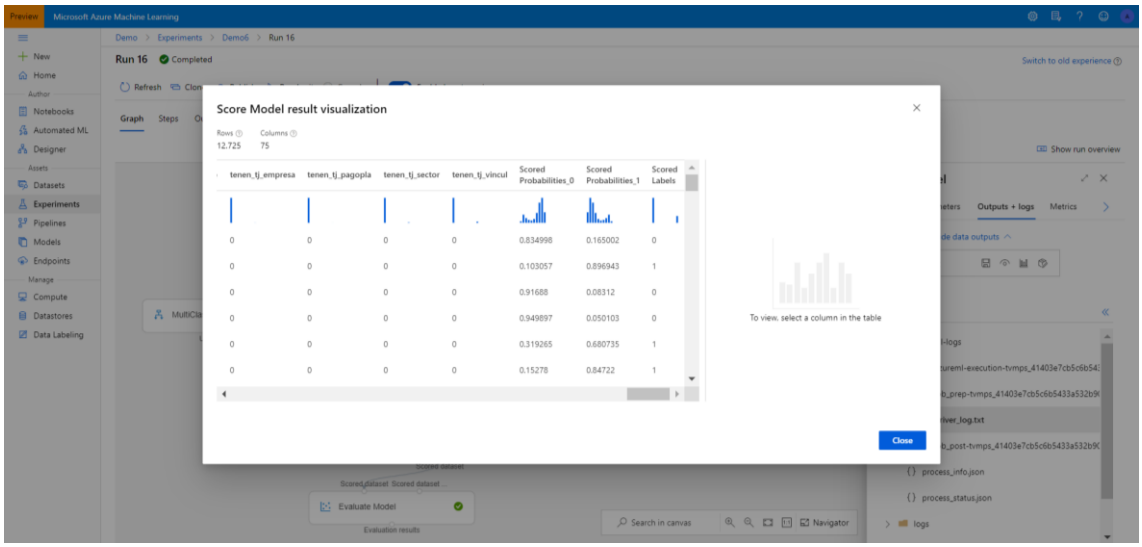


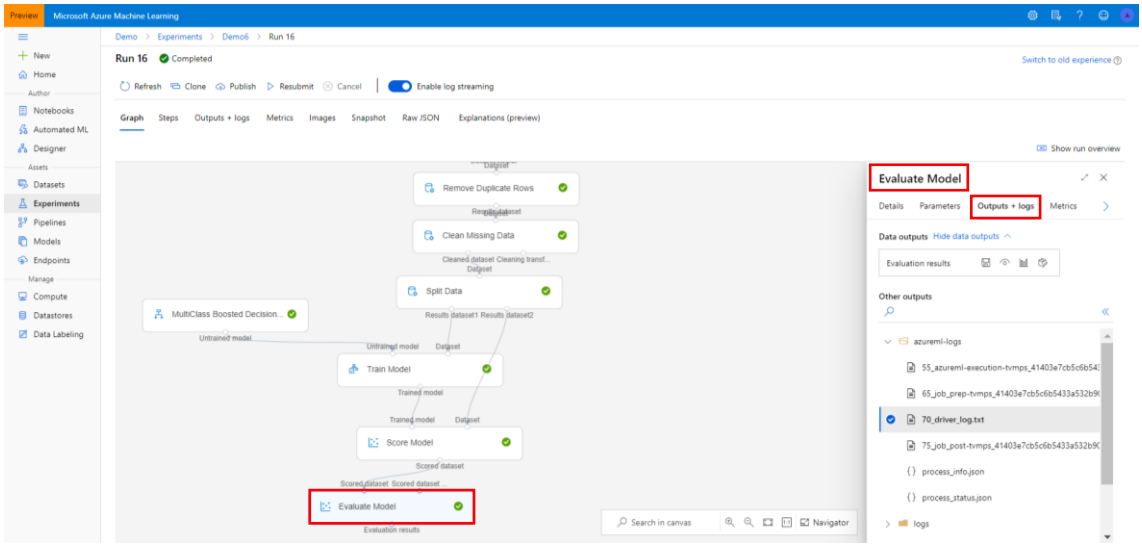
Por último se pulsa el botón "Submit" para poder acceder a los resultados obtenidos.



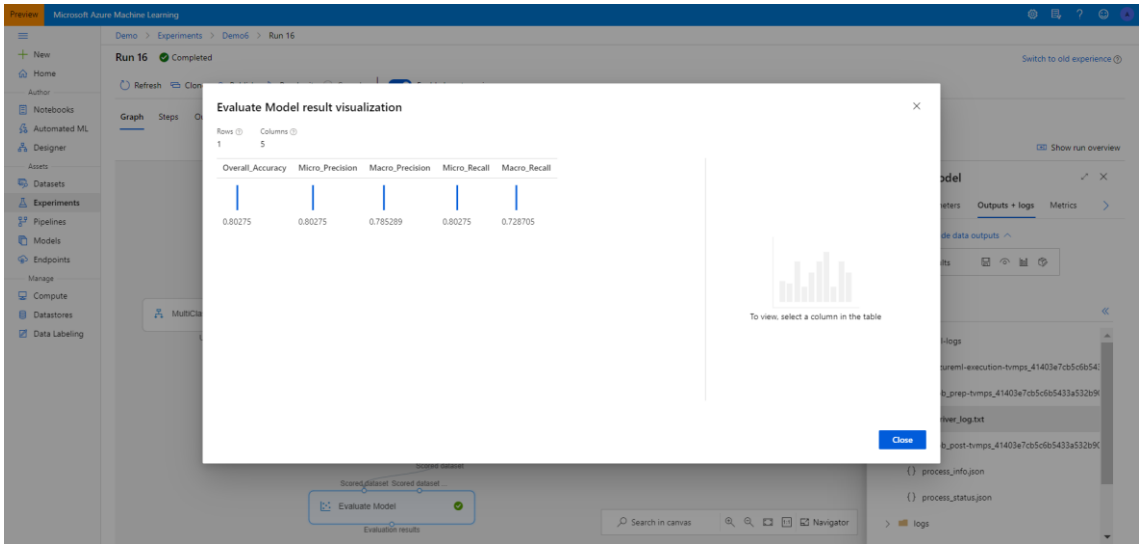
Los resultados obtenidos se pueden ver seleccionando el paso de "Score Model" y/o "Evaluate Model", y a continuación yendo a la salida de la derecha a la parte de "Outputs+Logs", en la parte de "Scored dataset", y pulsando en el tercer botón que aparece.


Microsoft Azure Machine Learning
Demo > Experiments > Demo6 > Run 16
Run 16 Completed
Refresh Clone Publish Resubmit Cancel
Enable log streaming
Graph Steps Outputs + logs Metrics Images Snapshot Raw JSON Explanations (preview)
Remove Duplicate Rows
Clean Missing Data
Split Data
MultiClass Boosted Decision...
Train Model
Score Model
Evaluate Model
Score Model
Details Parameters Outputs + logs Metrics
Data outputs
Scored dataset
Other outputs
azureml-logs
70_driver_log.txt
process_info.json
process_status.json
logs


Score Model result visualization
Rows 12.725
Columns 75
tenen_tj_empresa
tenen_tj_pagopla
tenen_tj_sector
tenen_tj_vincul
Scored Probabilities_0
Scored Probabilities_1
Scored Labels
To view, select a column in the table
Close



Los resultados de las métricas de evaluación del modelo son las siguientes:



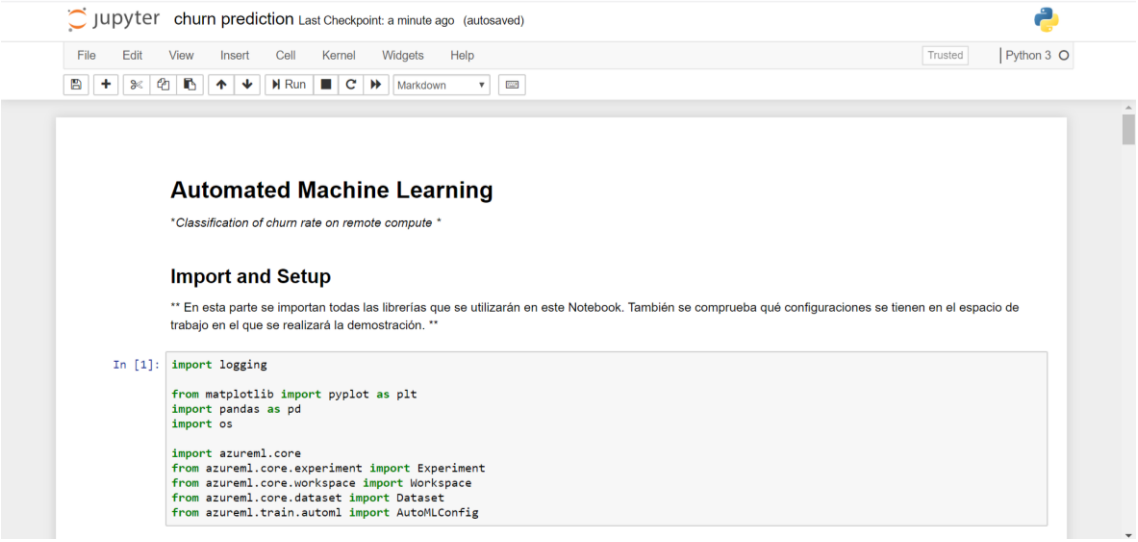
8. EJEMPLO 3

En esta última demostración se van a utilizar los cuadernos de Jupyter Notebook de Azure para el entrenamiento e implementación de modelos de Machine Learning, utilizando los SDK para Python en este caso. También existe la posibilidad de hacerlo utilizando los SDK para R.

En este punto, me parece interesante recalcar que Azure ML permite la combinación de diferentes opciones para solucionar un problema de Machine Learning concreto. Por ello, con el objetivo de mostrar esta característica de Azure ML, se ha optado por unir dos funcionalidades distintas como son los notebooks de Python (utilizando los SDK para Python) y la opción de Automated Machine Learning.

Para este ejemplo, se utiliza el conjunto de datos de "churn" para mostrar cómo se puede usar AutoML para un problema de clasificación simple. El objetivo es predecir si un cliente de un banco lo abandonará, o en su defecto no lo hará.

El cuaderno de Python con el script que se usa para esta demostración está utilizando computación remota para el entrenamiento del modelo.



jupyter churn prediction Last Checkpoint: 2 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

** Aquí lo único que se hace es ver qué versión de Azure ML SDK (1.5.0) se tiene, así como la versión que se utilizará en esta demostración (1.4.0). **

In [2]:
```
print("This notebook was created using version 1.5.0 of the Azure ML SDK")
print("You are currently using version", azureml.core.VERSION, "of the Azure ML SDK")
```

```
This notebook was created using version 1.5.0 of the Azure ML SDK
You are currently using version 1.4.0 of the Azure ML SDK
```

** Lo siguiente que se comprueba son diferentes especificaciones de la configuración del "Workspace" en el que se está trabajando (Subscription ID, Workspace, Resource Group, etc). **

jupyter churn prediction Last Checkpoint: 2 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [4]:
```
ws = Workspace.from_config()

# choose a name for experiment
experiment_name = 'automl-classification-churn'

experiment=Experiment(ws, experiment_name)

output = {}
output['Subscription ID'] = ws.subscription_id
output['Workspace'] = ws.name
output['Resource Group'] = ws.resource_group
output['Location'] = ws.location
output['Experiment Name'] = experiment.name
pd.set_option('display.max_colwidth', -1)
outputDf = pd.DataFrame(data = output, index = ['''])
outputDf.T
```

Out[4]:

Subscription ID	aa0731a9-87cf-48d3-99e7-1a19c2ba830c
Workspace	Demo
Resource Group	AzureML-Patricia
Location	westeurope
Experiment Name	automl-classification-churn

jupyter churn prediction Last Checkpoint: 5 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Create or Attach existing AmlCompute

** En esta parte básicamente lo que se hace es crear o utilizar un "Compute target" que ya existe, para ejecutar el run de Automated Machine Learning. En este caso se encuentra un "Compute target" (que ha sido configurado previamente), por lo que será el que se use para esta demostración. **

In [5]:
```
from azureml.core.compute import ComputeTarget, AmlCompute
from azureml.core.compute_target import ComputeTargetException

# Choose a name for your CPU cluster
cpu_cluster_name = "cpu-cluster-1"

# Verify that cluster does not exist already
try:
    compute_target = ComputeTarget(workspace=ws, name=cpu_cluster_name)
    print('Found existing cluster, use it.')
except ComputeTargetException:
    compute_config = AmlCompute.provisioning_configuration(vm_size='STANDARD_DS12_V2',
                                                            max_nodes=6)
    compute_target = ComputeTarget.create(ws, cpu_cluster_name, compute_config)

compute_target.wait_for_completion(show_output=True)
```

```
Found existing cluster, use it.
Succeeded
AmlCompute wait for completion finished

Minimum number of nodes requested have been provisioned
```

jupyter churn prediction Last Checkpoint: 7 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Data

Load Data

** Después se carga el dataset desde un archivo csv (datafinal.csv), que contiene tanto features como training labels. Las "features" son entradas al modelo, mientras que las "training labels" representan la salida esperada del modelo. A continuación, se dividen los datos usando random_split y se extraen los datos de entrenamiento para el modelo. **

In [6]:
```
from azureml.core import Workspace, Datastore, Dataset

datastore_name = 'churn'

# get existing workspace
workspace = Workspace.from_config()

# retrieve an existing datastore in the workspace by name
datastore = Datastore.get(workspace, datastore_name)
# create a TabularDataset from 3 file paths in datastore
datastore_paths = [(datastore, 'UI/05-18-2020_073016_UTC/datafinal.csv')]
churn_ds = Dataset.Tabular.from_delimited_files(path=datastore_paths)
data = churn_ds.to_pandas_dataframe()
```

jupyter churn prediction Last Checkpoint: 8 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
churn_ds = Dataset.Tabular.from_delimited_files(path=datastore_paths)
data = churn_ds.to_pandas_dataframe()
```

In [7]:
```
data.head(5)
```

Out[7]:

	churn	nacionalidad	renta_estimada	inprod	inserv	margen_mes	tenen_vista	tenen_ahpat	tenen_plazo	tenen_fondo	...	tenen_fconsumo	tenen_faapp	t
0	1	100	2	0	4	-1.19	1	0	0	0	...	0	0	
1	0	100	1	0	0	0.00	1	0	0	0	...	0	0	
2	1	100	2	6	7	79.11	1	0	0	0	...	0	0	
3	0	100	1	0	0	0.00	1	0	0	0	...	0	0	
4	0	100	3	2	6	18.25	1	0	0	0	...	0	0	

5 rows × 25 columns

In [9]:
```
training_data, validation_data = churn_ds.random_split(percentage=0.7, seed=223)
label_column_name = 'churn'
```

jupyter churn prediction Last Checkpoint: 9 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
label_column_name = 'churn'
```

Train

En esta parte se instancia un objeto AutoMLConfig. Esto define la configuración y los datos utilizados para ejecutar el experimento.

En la siguiente tabla se pueden observar algunas de las opciones que se tienen para la configuración del objeto.

Property	Description
task	classification or regression
primary_metric	This is the metric that you want to optimize. Classification supports the following primary metrics: *accuracy* *AUC_weighted* *average_precision_score_weighted* *norm_macro_recall* *precision_score_weighted*
enable_early_stopping	Stop the run if the metric score is not showing improvement.
n_cross_validations	Number of cross validation splits.
training_data	Input dataset, containing both features and label column.
label_column_name	The name of the label column.

Primary metrics

jupyter churn prediction Last Checkpoint: 10 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

Primary metrics

In [10]:
```
automl_settings = {
    "n_cross_validations": 3,
    "primary_metric": 'average_precision_score_weighted',
    "enable_early_stopping": True,
    "max_concurrent_iterations": 2,
    "experiment_timeout_hours": 0.25,
    "verbosity": logging.INFO,
}

automl_config = AutoMLConfig(task = 'classification',
                             debug_log = 'automl_errors.log',
                             compute_target = compute_target,
                             training_data = training_data,
                             label_column_name = label_column_name,
                             **automl_settings
                            )
```

** Llame al método "submit" en el objeto de experimento y pase la configuración de ejecución. **

In [11]:
```
remote_run = experiment.submit(automl_config, show_output = False)
```

```
Running on remote or ADB.
```

jupyter churn prediction Last Checkpoint: 11 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

** Como se hace a continuación, se puede obtener un link a Azure Machine Learning Studio, así como un link a la documentación de Azure Machine Learning SDK para Python. **

In [12]:
```
remote_run
```

Out[12]:

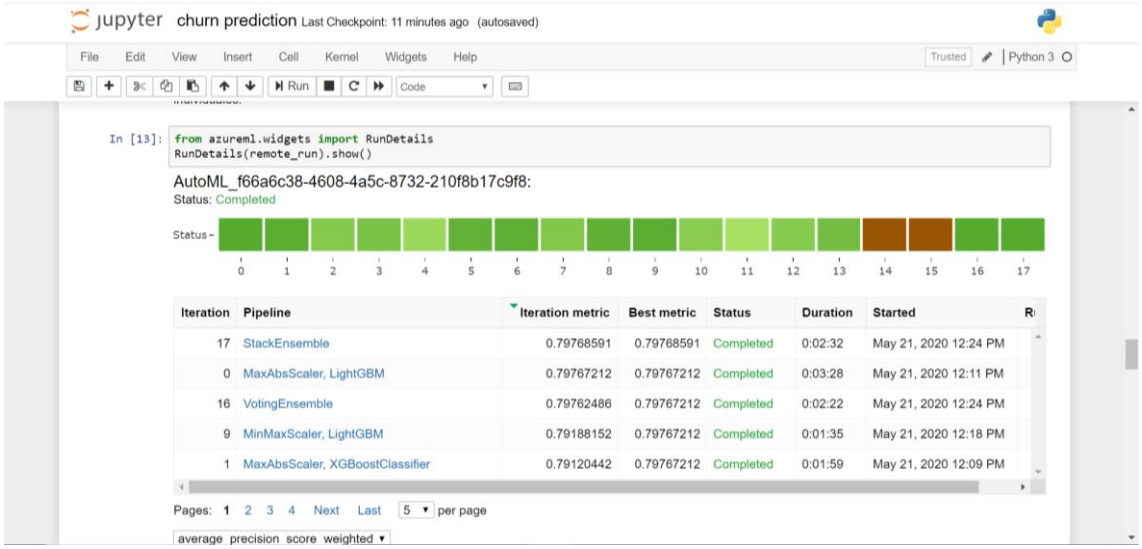
Experiment	Id	Type	Status	Details Page	Docs Page
automl-classification-churn	AutoML_f66a6c38-4608-4a5c-8732-210f8b17c9f8	automl	NotStarted	Link to Azure Machine Learning studio	Link to Documentation

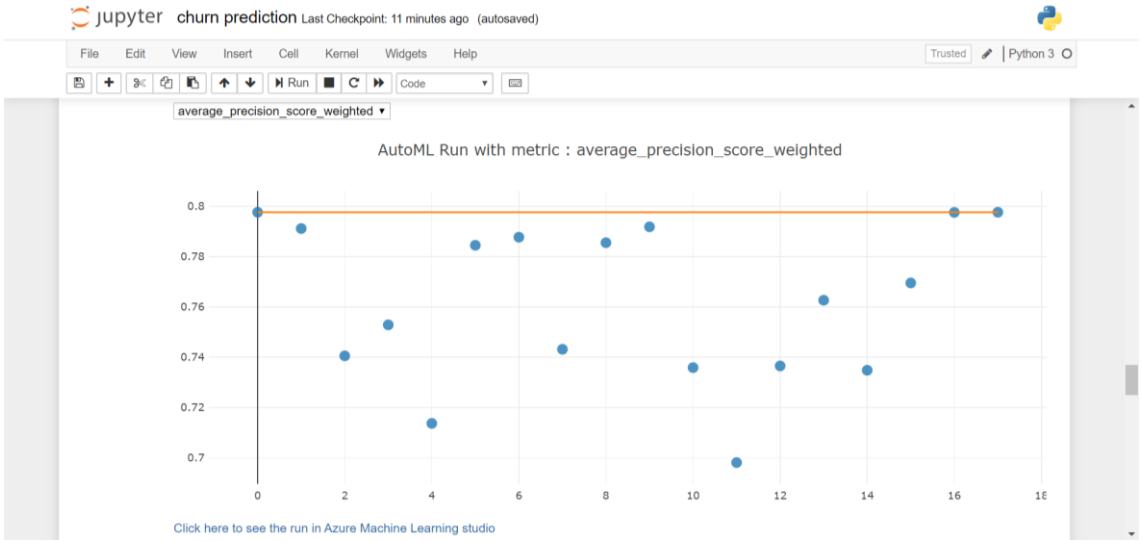
Results

Widget for Monitoring Runs

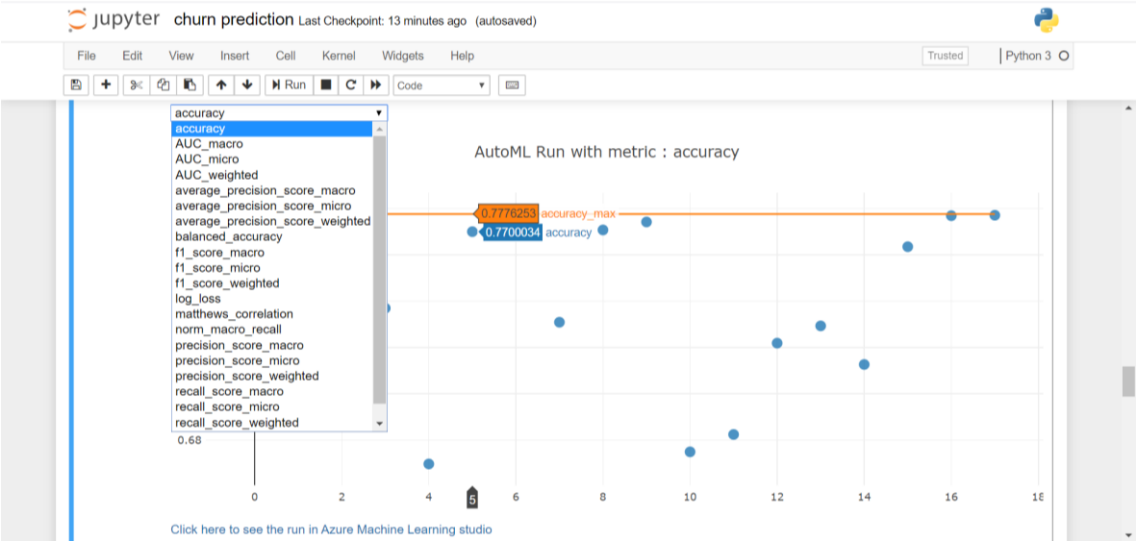
** El widget informará primero sobre el estado de "carga" mientras ejecuta la primera iteración. Después de completar la primera iteración, se mostrará un gráfico y una tabla que se actualiza de manera automática. El widget se actualizará una vez por minuto, por lo que debería ver la actualización del gráfico a medida que se ejecutan las ejecuciones secundarias. **

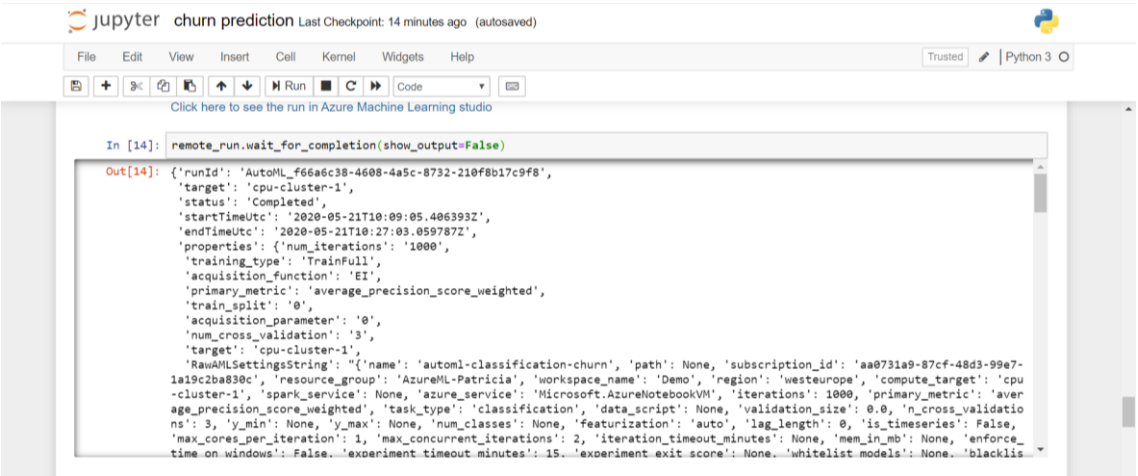
Nota: El widget muestra un enlace en la parte inferior. Si pulsa en este enlace se abrirá una interfaz web donde podrá explorar los detalles de ejecución individuales.


jupyter churn prediction Last Checkpoint: 11 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [13]:
from azureml.widgets import RunDetails
RunDetails(remote_run).show()
AutoML_f66a6c38-4608-4a5c-8732-210f8b17c9f8:
Status: Completed
Status -
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Iteration | Pipeline | Iteration metric | Best metric | Status | Duration | Started
17 | StackEnsemble | 0.79768591 | 0.79768591 | Completed | 0:02:32 | May 21, 2020 12:24 PM
0 | MaxAbsScaler, LightGBM | 0.79767212 | 0.79767212 | Completed | 0:03:28 | May 21, 2020 12:11 PM
16 | VotingEnsemble | 0.79762486 | 0.79767212 | Completed | 0:02:22 | May 21, 2020 12:24 PM
9 | MinMaxScaler, LightGBM | 0.79188152 | 0.79767212 | Completed | 0:01:35 | May 21, 2020 12:18 PM
1 | MaxAbsScaler, XGBoostClassifier | 0.79120442 | 0.79767212 | Completed | 0:01:59 | May 21, 2020 12:09 PM
Pages: 1 2 3 4 Next Last 5 per page
average_precision_score_weighted


jupyter churn prediction Last Checkpoint: 11 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
average_precision_score_weighted
AutoML Run with metric : average_precision_score_weighted
0.8
0.78
0.76
0.74
0.72
0.7
0 2 4 6 8 10 12 14 16 18
Click here to see the run in Azure Machine Learning studio

En la siguiente imagen se pueden ver las diferentes opciones que se tienen a la hora de visualizar la métrica que se quiera. En este caso se quiere ver la precisión o accuracy.





jupyter churn prediction Last Checkpoint: 15 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Analyze results

Retrieve the Best Model

** A continuación seleccionamos el mejor pipeline de todas las iteraciones que se han realizado. El método "get_output" devuelve la mejor ejecución y el modelo ajustado. **

In [15]:
```
best_run, fitted_model = remote_run.get_output()
fitted_model
```

Out[15]:
```
Pipeline(memory=None,
     steps=[('datatransformer', DataTransformer(enable_dnn=None, enable_feature_sweeping=None,
        feature_sweeping_config=None, feature_sweeping_timeout=None,
        featurization_config=None, force_text_dnn=None,
        is_cross_validation=None, is_onnx_compatible=None, logger=None,
        obser...7f8761f6cc18>,
           solver='lbfgs', tol=0.0001, verbose=0),
            training_cv_folds=5))])
```

Print the properties of the model

El "fitted_model" es un objeto de Python que puede usar para leer las diferentes propiedades del objeto.

jupyter churn prediction Last Checkpoint: a few seconds ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Test the fitted model

** Ahora que el modelo está entrenado, se dividen los datos de la misma manera que se dividieron los datos para el entrenamiento (la diferencia aquí es que los datos se dividen de forma local). Luego se ejecutan los datos de prueba a través del modelo entrenado para obtener los valores pronosticados. **

In [16]:
```
# convertir los datos del test en un dataframe
X_test_df = validation_data.drop_columns(columns=[label_column_name]).to_pandas_dataframe()
y_test_df = validation_data.keep_columns(columns=[label_column_name], validate=True).to_pandas_dataframe()
```

In [17]:
```
# llamar al método "predict" del modelo
y_pred = fitted_model.predict(X_test_df)
y_pred
```

Out[17]:
```
array([1, 1, 1, ..., 0, 0, 0])
```

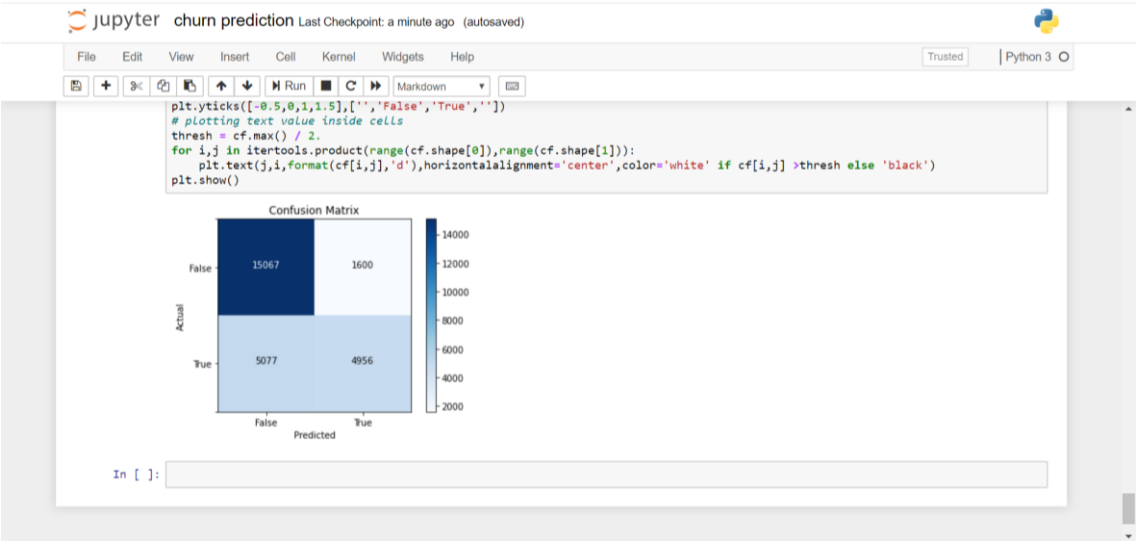
jupyter churn prediction Last Checkpoint: a few seconds ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

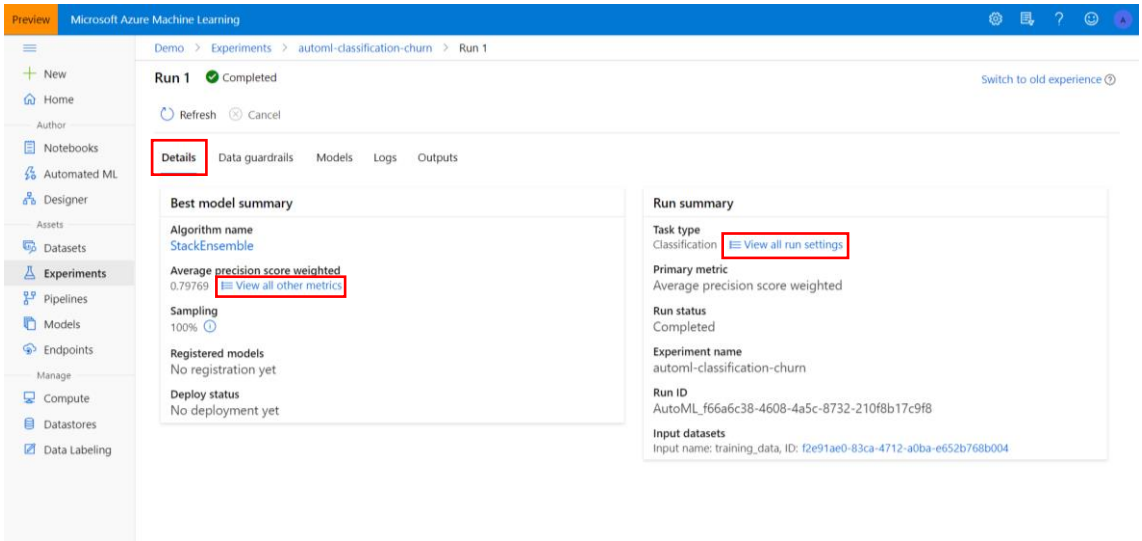
Calculate metrics for the prediction

** Finalmente se visualizan los datos en un diagrama de dispersión para mostrar cuáles son nuestros valores de verdad (reales) comparados con los valores predichos del modelo entrenado. **

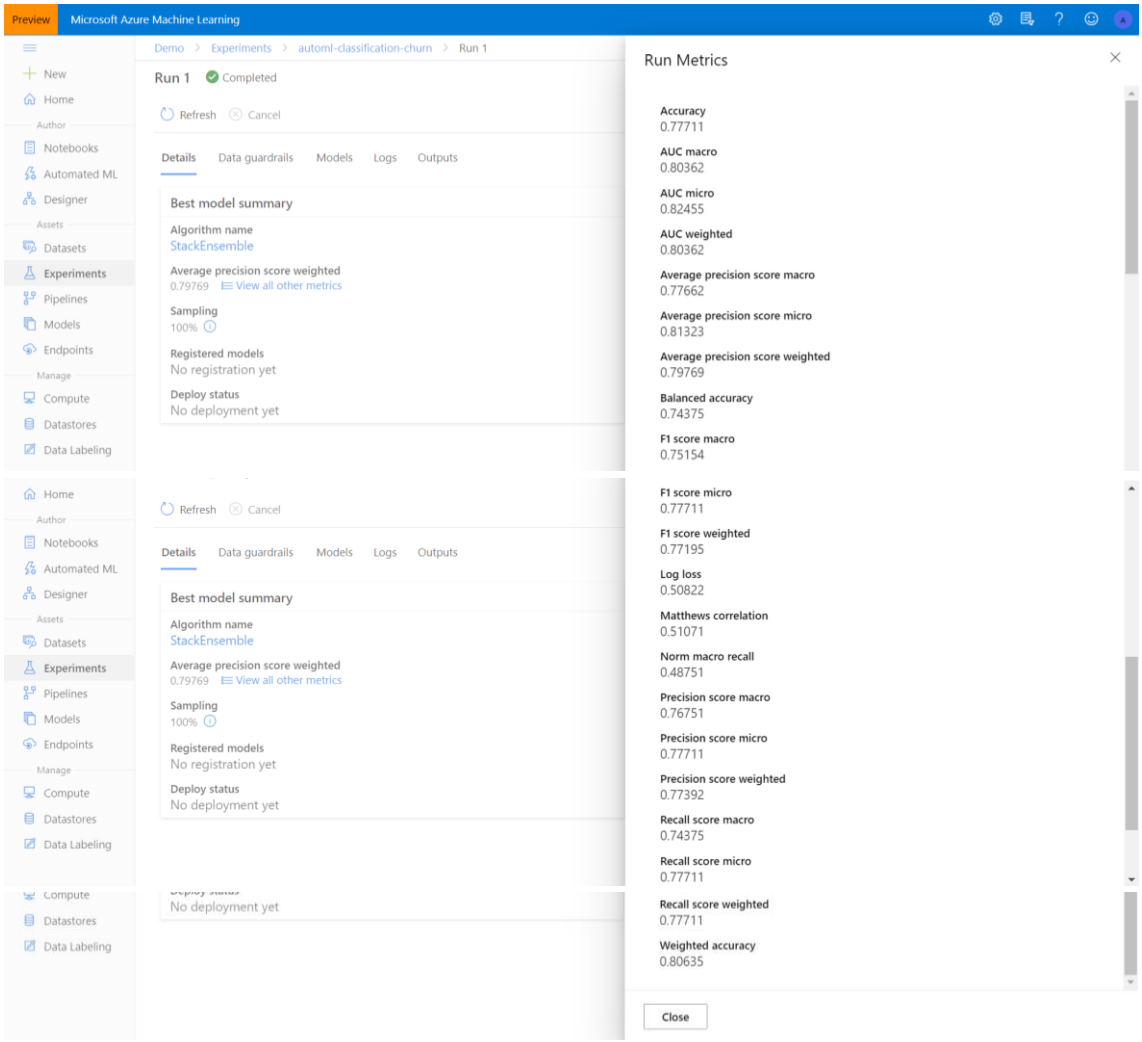
In [18]:
```
from sklearn.metrics import confusion_matrix
import numpy as np
import itertools

cf =confusion_matrix(y_test_df.values,y_pred)
plt.imshow(cf,cmap=plt.cm.Blues,interpolation='nearest')
plt.colorbar()
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
class_labels = ['False','True']
tick_marks = np.arange(len(class_labels))
plt.xticks(tick_marks,class_labels)
plt.yticks([-0.5,0,1,1.5],['','False','True',''])
# plotting text value inside cells
thresh = cf.max() / 2.
for i,j in itertools.product(range(cf.shape[0]),range(cf.shape[1])):
    plt.text(j,i,format(cf[i,j],'d'),horizontalalignment='center',color='white' if cf[i,j] >thresh else 'black')
plt.show()
```



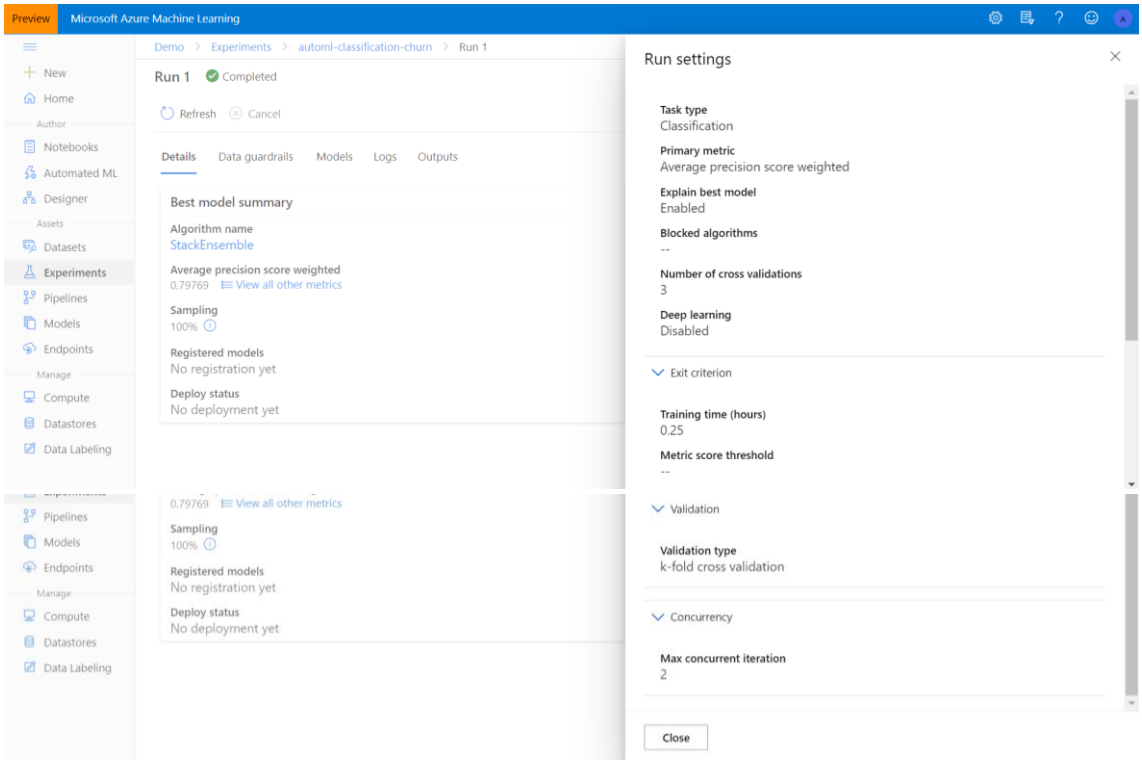
A continuación, si se quiere tener mayor nivel de detalle de todo el proceso, hay que irse a Azure ML. En la parte de "Details" aparecen diferentes detalles del experimento que ha sido creado usando los Notebooks de Python. El mejor modelo conseguido ha sido utilizando el algoritmo de "StackEnsemble". Se ha conseguido un "Average precision score weighted" de 0.79769.



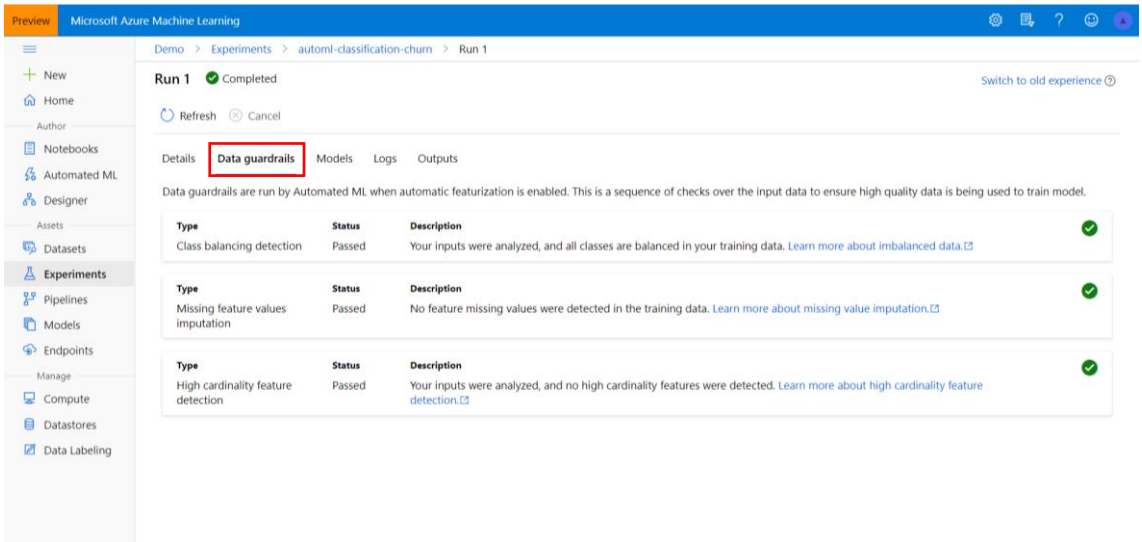
También se pueden consultar otras métricas. Para ello se debe pulsar en donde pone "View all other metrics".



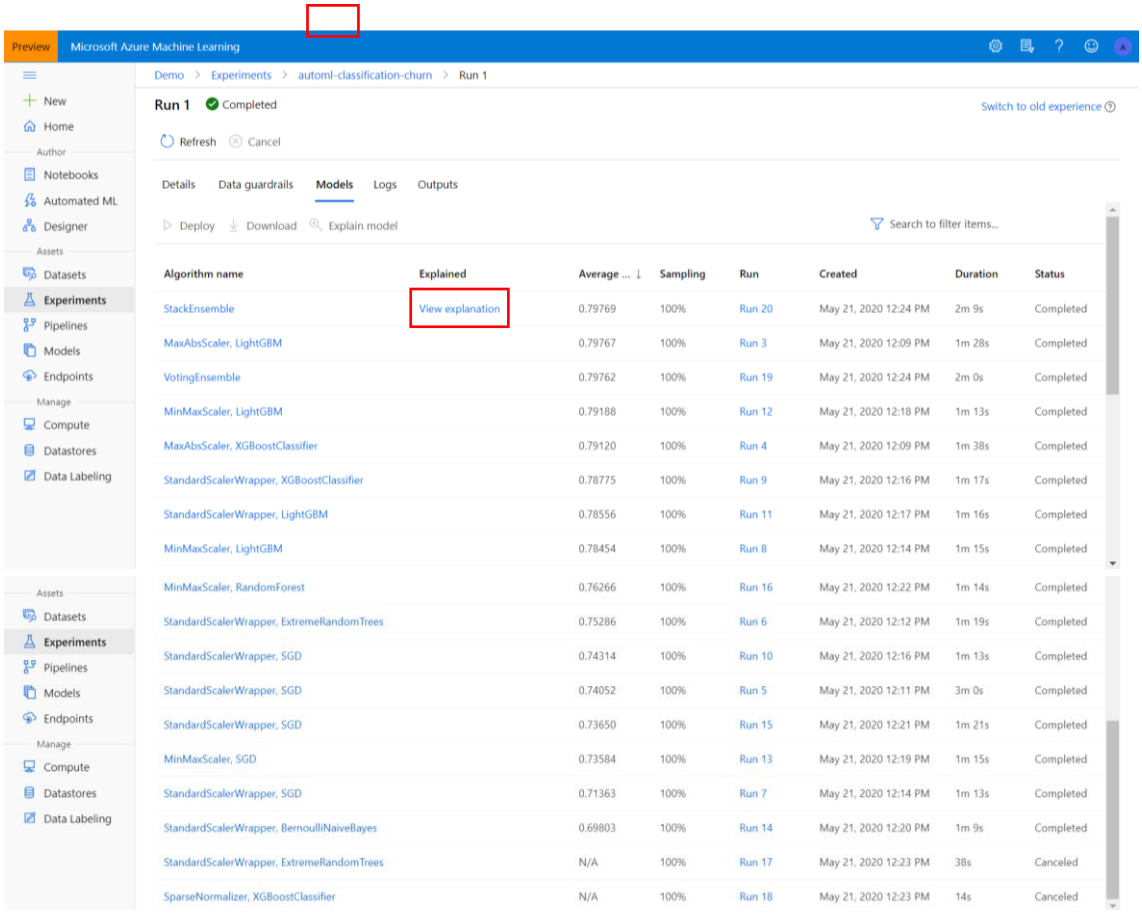
También se puede tener más detalle sobre la configuración utilizada para la creación del modelo. Para ello se debe pulsar en donde pone "View all run settings".



En la parte de "Data guardrails" se pueden ver una secuencia de verificaciones sobre los datos de entrada para garantizar que se utilicen datos de alta calidad para entrenar el modelo.

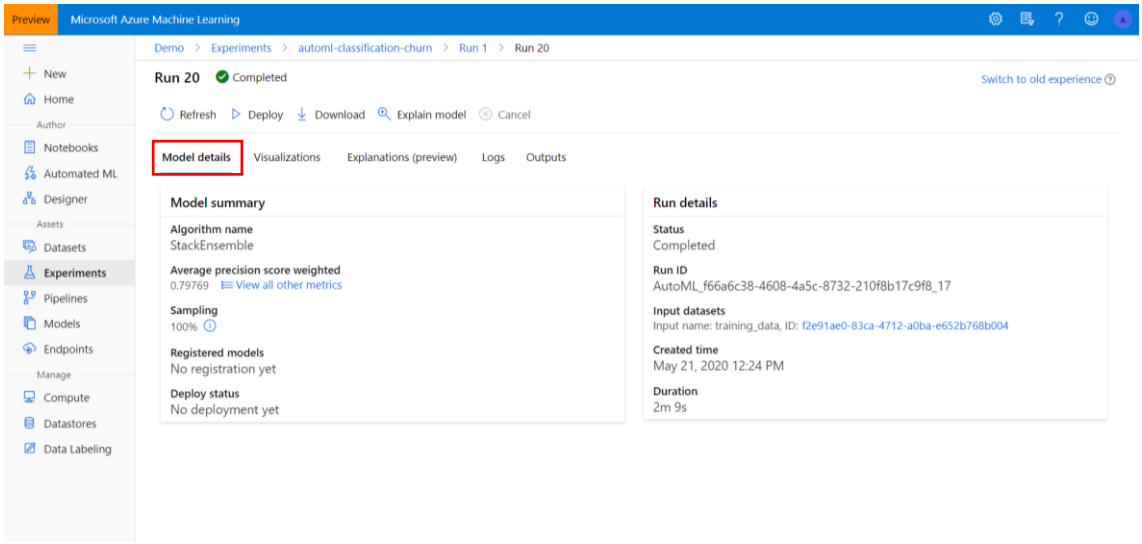


En la parte de "Models" aparecen todos los modelos que han sido probados en la ejecución para conseguir el mejor modelo, utilizando como métrica el "Average precision score weighted".

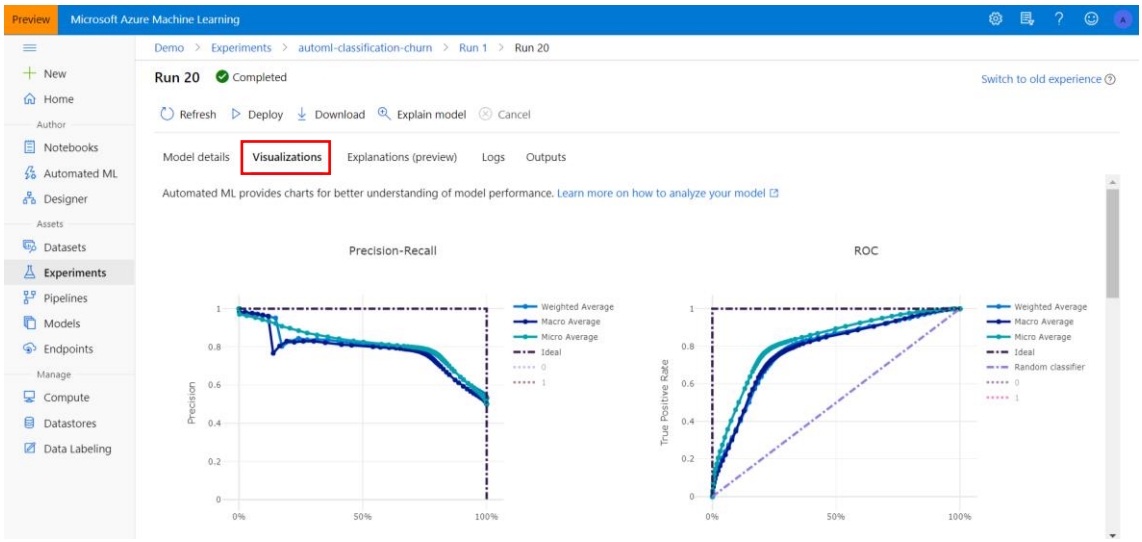


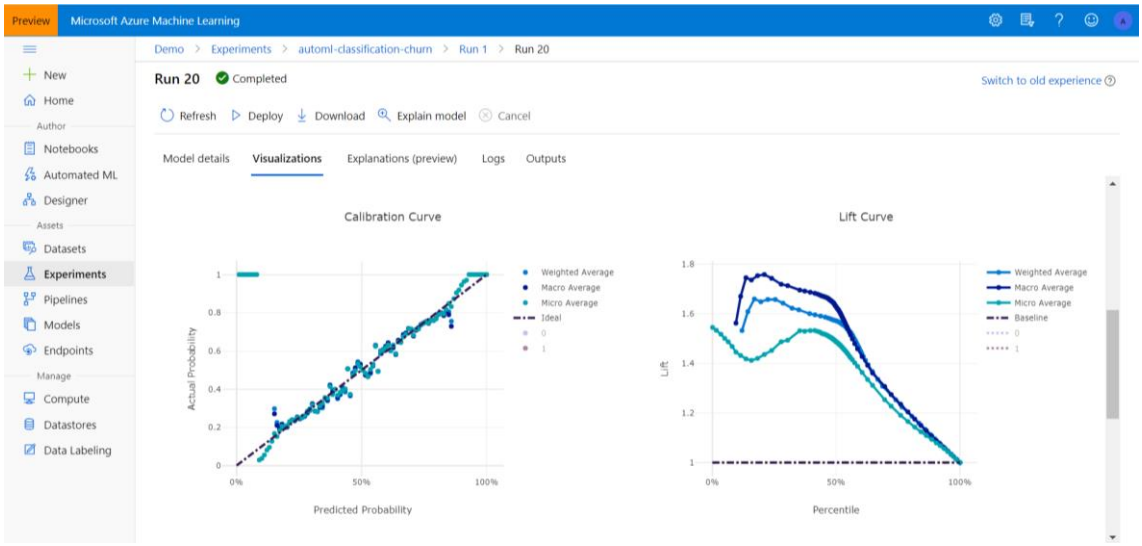
Si queremos más detalles sobre el mejor modelo, en este caso "StackEnsemble", pulse en donde pone "View explanation".

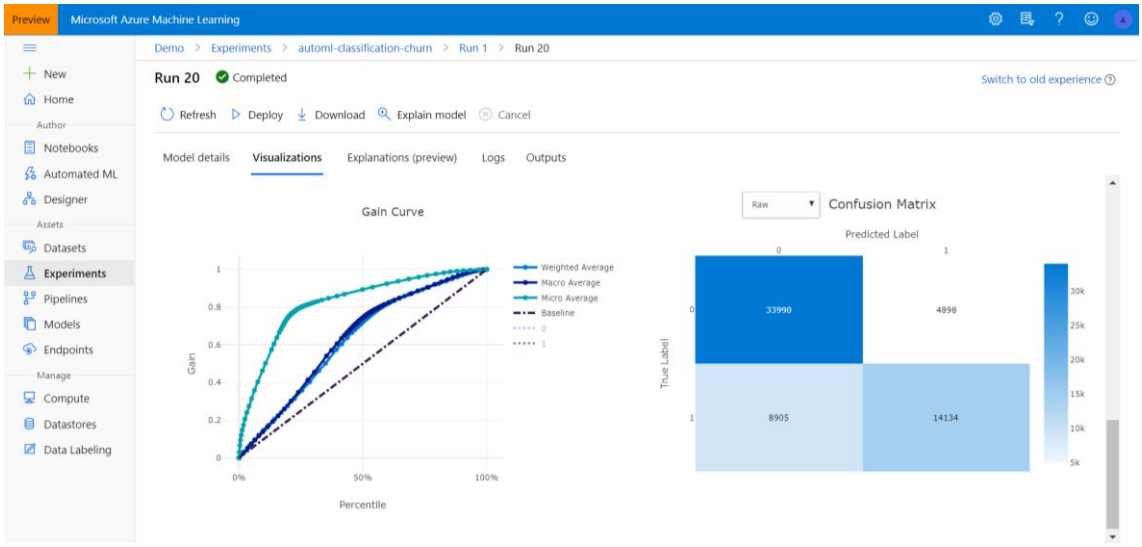
En la pestaña "Model details" aparecen los detalles del mejor modelo seleccionado. En este caso ha sido "Stack Ensemble" con un "Average precision score weighted" de 0.79769.

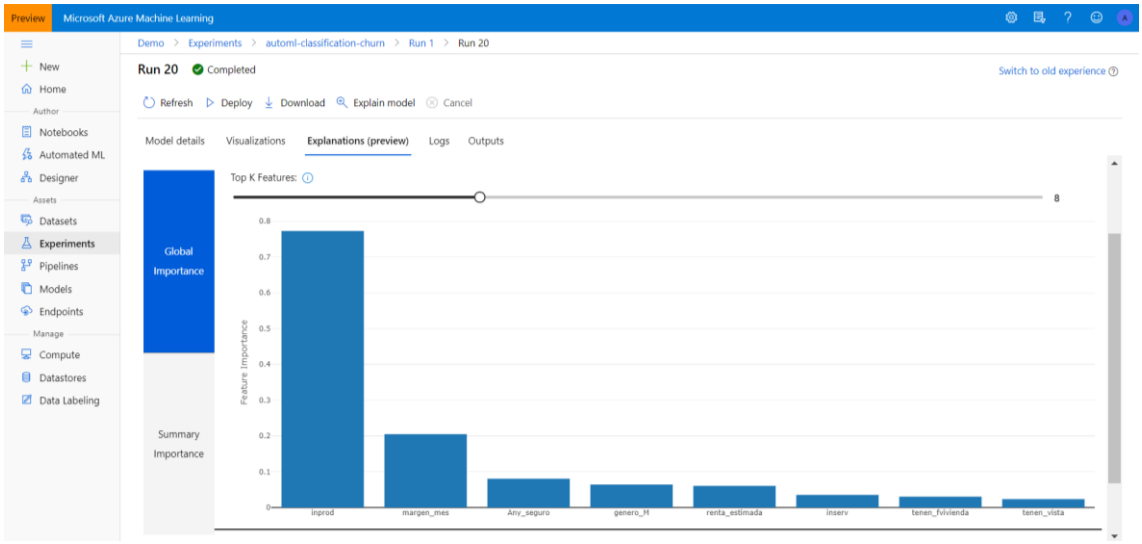


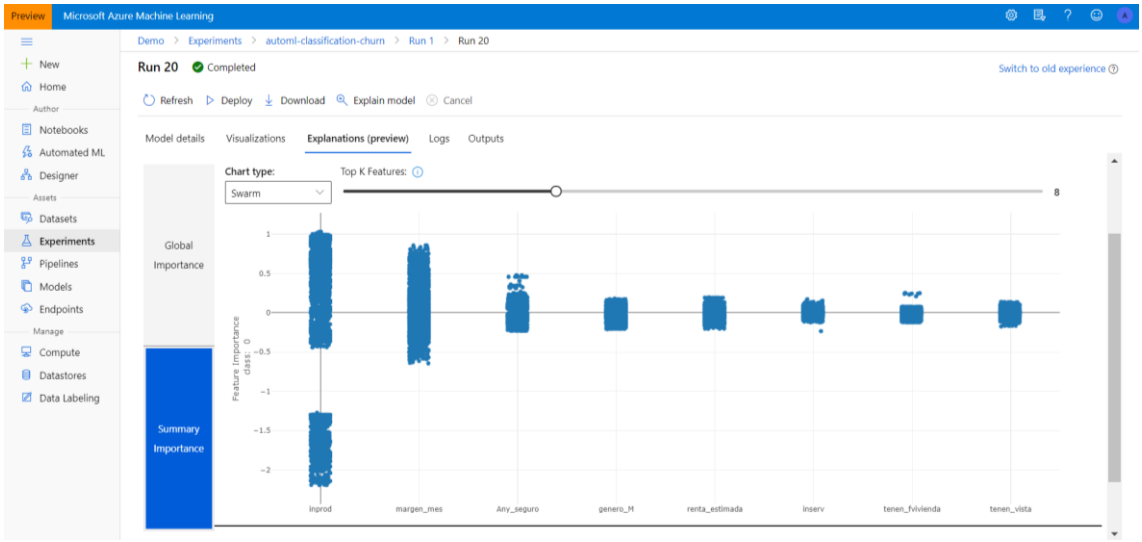
A continuación, en la pestaña de "Visualizations" aparecen diferentes gráficos de distintas métricas.




Preview
Microsoft Azure Machine Learning
Demo > Experiments > automl-classification-churn > Run 1 > Run 20
Run 20
Completed
Switch to old experience
Refresh
Deploy
Download
Explain model
Cancel
Model details
Visualizations
Explanations (preview)
Logs
Outputs
New
Home
Author
Notebooks
Automated ML
Designer
Assets
Datasets
Experiments
Pipelines
Models
Endpoints
Manage
Compute
Datastores
Data Labeling
Calibration Curve
Actual Probability
Predicted Probability
Weighted Average
Macro Average
Micro Average
Ideal
Lift Curve
Lift
Percentile
Baseline


Preview
Microsoft Azure Machine Learning
Demo > Experiments > automl-classification-churn > Run 1 > Run 20
Run 20
Completed
Switch to old experience
Refresh
Deploy
Download
Explain model
Cancel
Model details
Visualizations
Explanations (preview)
Logs
Outputs
New
Home
Author
Notebooks
Automated ML
Designer
Assets
Datasets
Experiments
Pipelines
Models
Endpoints
Manage
Compute
Datastores
Data Labeling
Gain Curve
Gain
Percentile
Weighted Average
Macro Average
Micro Average
Baseline
Raw
Confusion Matrix
Predicted Label
True Label
33990
4898
8905
14134


Preview
Microsoft Azure Machine Learning
Demo > Experiments > automl-classification-churn > Run 1 > Run 20
Run 20
Completed
Switch to old experience
Refresh
Deploy
Download
Explain model
Cancel
Model details
Visualizations
Explanations (preview)
Logs
Outputs
Top K Features:
8
Global Importance
Summary Importance
Feature Importance
inprod
margen_mes
Any_seguro
genero_M
renta_estimada
inserv
tenen_fvivienda
tenen_vista


Preview
Microsoft Azure Machine Learning
Demo > Experiments > automl-classification-churn > Run 1 > Run 20
Run 20
Completed
Switch to old experience
Refresh
Deploy
Download
Explain model
Cancel
Model details
Visualizations
Explanations (preview)
Logs
Outputs
Chart type:
Swarm
Top K Features:
8
Global Importance
Summary Importance
Feature Importance
inprod
margen_mes
Any_seguro
genero_M
renta_estimada
inserv
tenen_fvivienda
tenen_vista

9. CONCLUSIONES

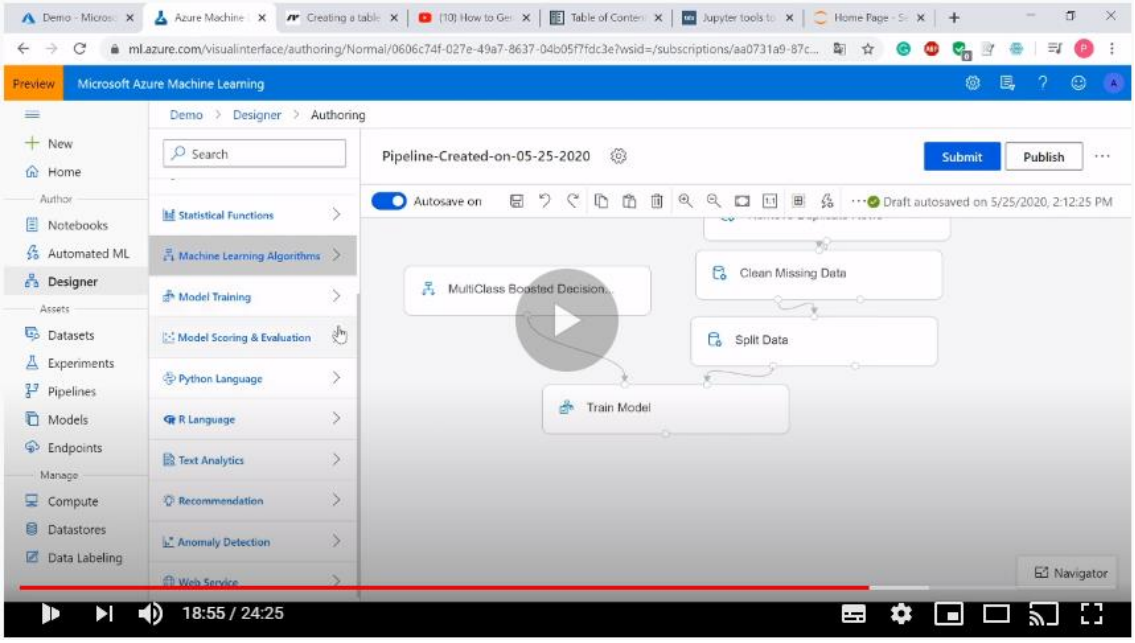
Hoy en día uno de los temas de moda es el término de **Machine Learning**. A pesar de su creciente demanda, hoy en día presente en casi todas las conversaciones a nivel empresarial, realmente siempre ha estado en nuestras vidas, y desde hace mucho tiempo. Lo que antes se llamaba "Data Mining", hoy ha pasado a llamarse "Machine Learning". De hecho, hay bastante falta de información en cuanto a la terminología existente. Cuando se habla de Machine Learning, se habla de una de las ramas que hay dentro de la Inteligencia Artificial.

Pero lo que no todos saben es las limitaciones de las soluciones del Machine Learning tradicional. Altos costes en términos de capacidad de cómputo y almacenamiento, los datos están restringidos en muchas ocasiones, las herramientas existentes suelen ser complejas y estar fragmentadas, ello hace que la colaboración se vea limitada, pero el mayor problema de todos es la puesta en producción de un modelo de Machine Learning, por lo que, en muchos casos, nunca se llegan a implementar.

Precisamente, para solucionar todas estas barreras de entrada a la hora de implementar un modelo de Machine Learning en una empresa, aparece Azure ML. Una herramienta muy potente como se ha demostrado en este trabajo, que si se integra con los recursos tradicionales de "hacer" Machine Learning, puede dar lugar a una reducción considerable de costes, aumento de la productividad, y como consecuencia, al éxito de un proyecto de Machine Learning.

10. VIDEOTUTORIAL

En el siguiente Videotutorial mostramos una Introducción a Microsoft Azure Machine Learning. Una completa descripción de todos los componentes, con ejemplos prácticos sobre cómo usar una de las plataformas Cloud Analytics más potentes y completas en la actualidad



11. PROBLEMAS ENCONTRADOS

Algunas de las dificultades encontradas en el desarrollo de este proyecto han sido:

- Mucha falta de documentación ejemplificada. La mayoría de los videotutoriales y ejemplos se hacen utilizando la interfaz de Azure ML Studio. Apenas hay entradas en páginas web o blogs sobre cómo empezar a trabajar en Azure ML.
- Es difícil para una persona que nunca ha trabajado con Azure ML poder utilizar Azure Calculator, no es nada intuitivo si no se sabe las características ni especificaciones de cada máquina virtual. También es complicado seleccionar el "Compute target" más adecuado para tú proyecto, etc.
- Cuando tienes un problema tienes que abrir un "Report" y un caso con tú problema en concreto. Tarda bastante en ser resuelto, e incluso muchas veces no te solucionan el problema.
- No es posible borrar experimentos que hayas hecho y que ya no te sean útiles.
- El tema de las regiones y las "quotas" tampoco es muy intuitivo.
- Tampoco es fácil saber cuál es la ruta del dataset o datastore que estés utilizando. Por ejemplo:

PROBLEMA:

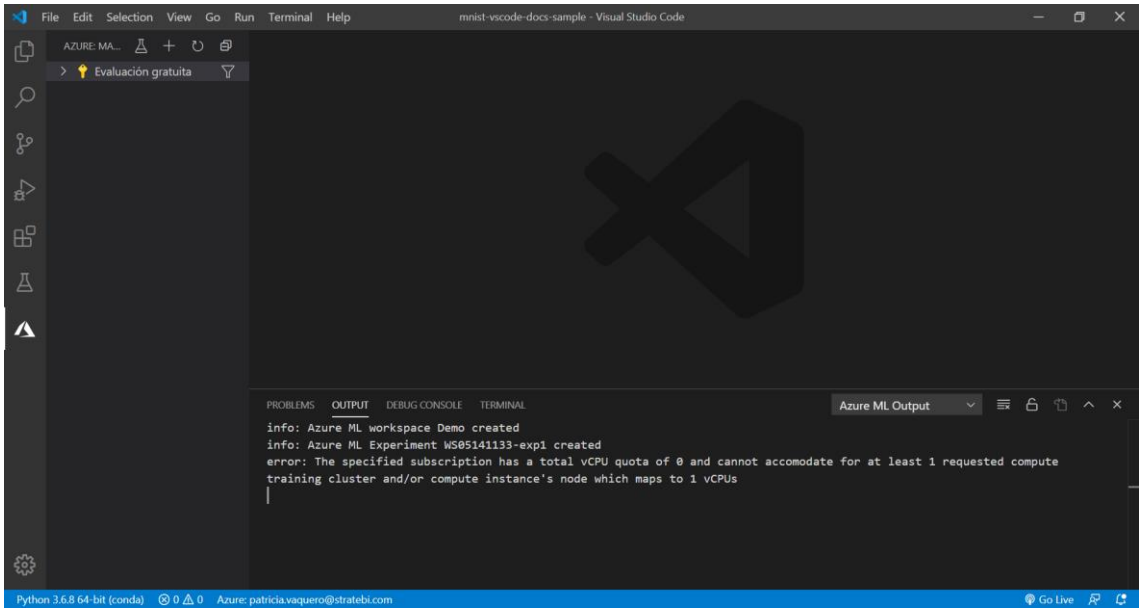
data = pd.read_csv("https://automlsamplenotebookdata.blob.core.windows.net/automl-sample-notebook-data/bankmarketing_train.csv")

SOLUCIÓN:

test_dataset = Dataset.Tabular.from_delimited_files(path = [(datastore, blobstore_datadir + '/test_data.csv')])

- En general la documentación oficial está bien, pero aun así tiene carencias, como por ejemplo en toda la parte de cómo "levantar" las máquinas virtuales, así como explicaciones exhaustivas de las diferentes características de cada máquina virtual, etc.

- Con la Evaluación gratuita se dispone de vCPU quota de 0, con lo que realmente no se puede hacer nada en Azure ML sin pasarse a la opción de pago. Ejemplo:



- La MEJOR CUOTA es la de “Standard FSv2 Family vCPU’s”.

12. BIBLIOGRAFÍA

https://docs.microsoft.com/en-us/azure/machine-learning/

https://azure.microsoft.com/en-us/overview/ai-platform/dev-resources/?&WT.mc_id=aishow-c9-sejuare#resources-explore

https://www.blue-granite.com/blog/train-and-deploy-machine-learning-models-using-the-azureml-service

https://azure.microsoft.com/es-es/overview/what-is-the-cloud/

https://medium.com/@experienclA18/diferencias-entre-la-inteligencia-artificial-y-el-machine-learning-f0448c503cd4

https://en.wikipedia.org/wiki/Platform_as_a_service

https://blogs.solidq.com/es/business-analytics/un-paseo-por-azure-ml-services/

https://weekly-geekly-es.github.io/articles/485338/index.html

https://github.com/MicrosoftDocs/azure-docs.es-es/blob/master/articles/machine-learning/overview-what-is-azure-ml.md

https://github.com/Azure/azure-sdk-for-python

https://github.com/Azure/MachineLearningNotebooks

https://azure.github.io/azure-sdk-for-python/

https://azure.github.io/azure-sdk/releases/latest/all/dotnet.html

https://www.youtube.com/watch?v=JqVibiT3Uuk

https://channel9.msdn.com/Shows/AI-Show/Allup-Azure-ML

https://www.paradigmadigital.com/dev/comparativa-servicios-cloud-aws-azure-gcp/

https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

https://es.wikipedia.org/wiki/Matriz_de_confusi%C3%B3n

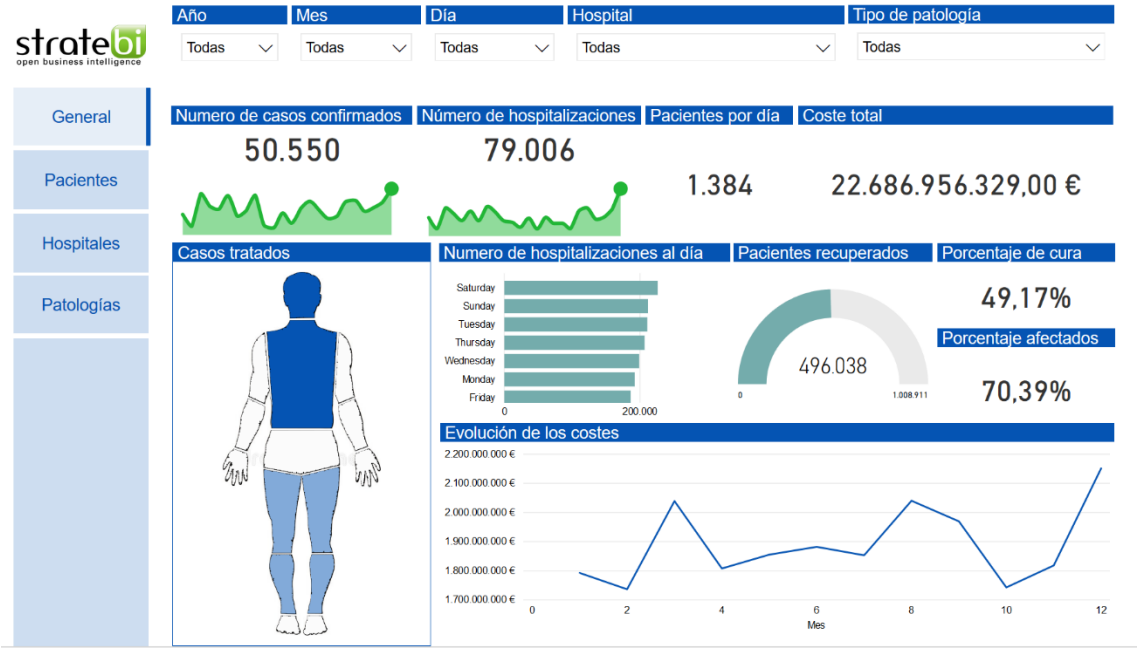
https://es.wikipedia.org/wiki/Curva_ROC

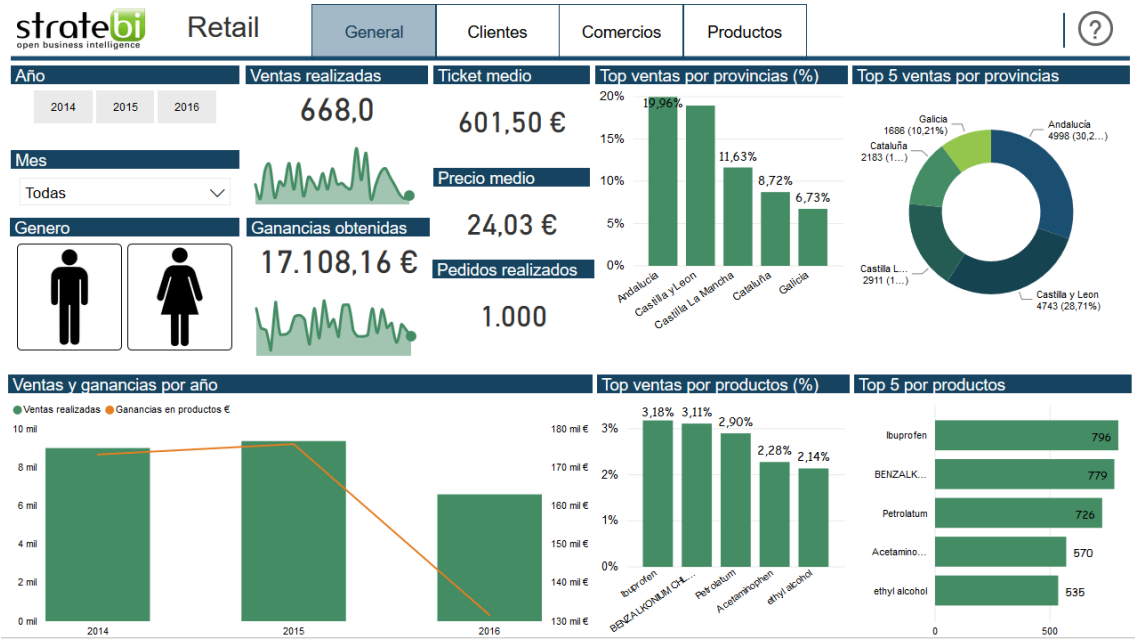
https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/boosted-decision-tree-regression

https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

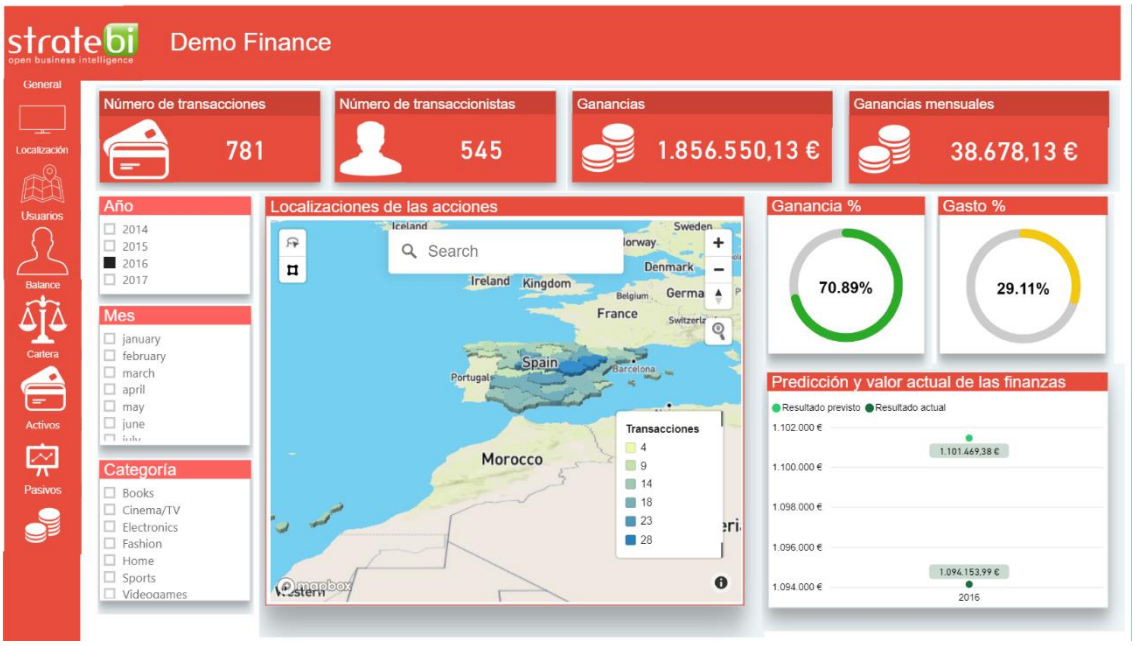
13. POWER BI

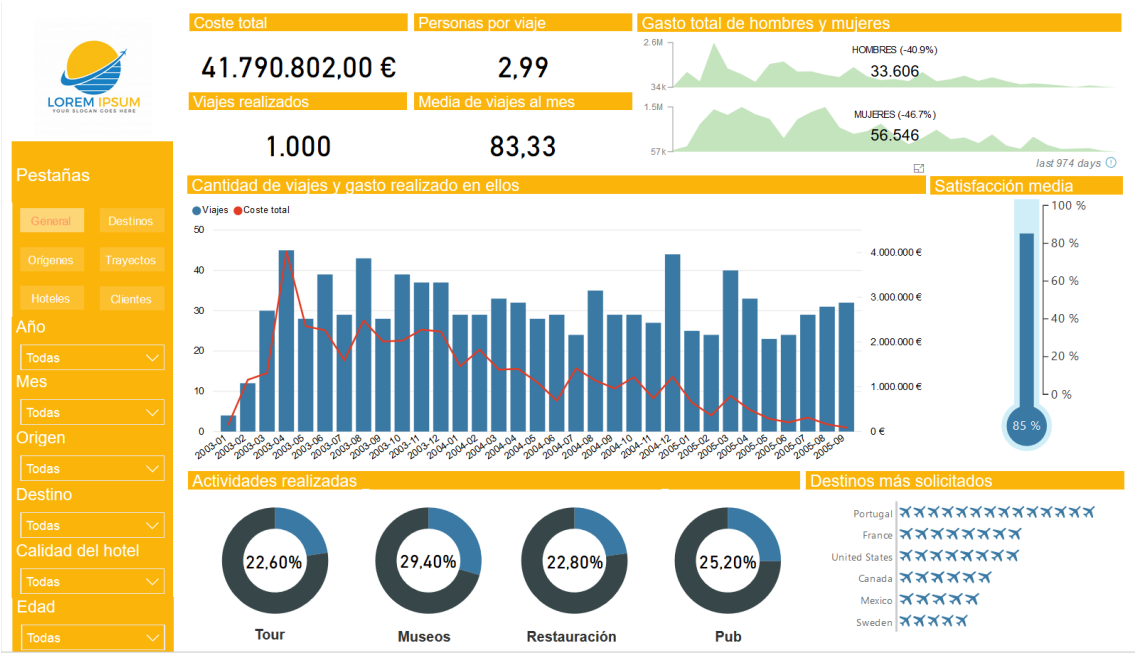
Stratebi es Partner Certificado en Microsoft Power BI. En esta sección puedes consultar algunas **Demos Online** en donde ver el potencial de la herramienta, así como algunos videotutoriales


Año
Mes
Día
Hospital
Tipo de patología
Todas
General
Pacientes
Hospitales
Patologías
Numero de casos confirmados
50.550
Número de hospitalizaciones
79.006
Pacientes por día
1.384
Coste total
22.686.956.329,00 €
Casos tratados
Numero de hospitalizaciones al día
Saturday
Sunday
Tuesday
Thursday
Wednesday
Monday
Friday
Pacientes recuperados
496.038
Porcentaje de cura
49,17%
Porcentaje afectados
70,39%
Evolución de los costes
Mes


Retail
General
Clientes
Comercios
Productos
Año
2014
2015
2016
Mes
Todas
Genero
Ventas realizadas
668,0
Ganancias obtenidas
17.108,16 €
Ticket medio
601,50 €
Precio medio
24,03 €
Pedidos realizados
1.000
Top ventas por provincias (%)
19,96%
11,63%
8,72%
6,73%
Andalucia
Castilla y Leon
Castilla La Mancha
Cataluña
Galicia
Top 5 ventas por provincias
Galicia 1686 (10,21%)
Andalucia 4998 (30,2...)
Cataluña 2183 (1...)
Castilla L... 2911 (1...)
Castilla y Leon 4743 (28,71%)
Ventas y ganancias por año
Ventas realizadas
Ganancias en productos €
Top ventas por productos (%)
3,18%
3,11%
2,90%
2,28%
2,14%
Ibuprofen
BENZALKONIUM CHL...
Petrolatum
Acetaminophen
ethyl alcohol
Top 5 por productos
Ibuprofen 796
BENZALK... 779
Petrolatum 726
Acetamino... 570
ethyl alcohol 535


Year
2019
Financial Results - Graphs
stratebi open business intelligence
Earnings
Direct Expenses
Gross Margin
Commercial Expenses
Contribution Margin
EBITDA
Corporative EBITDA
EBIT

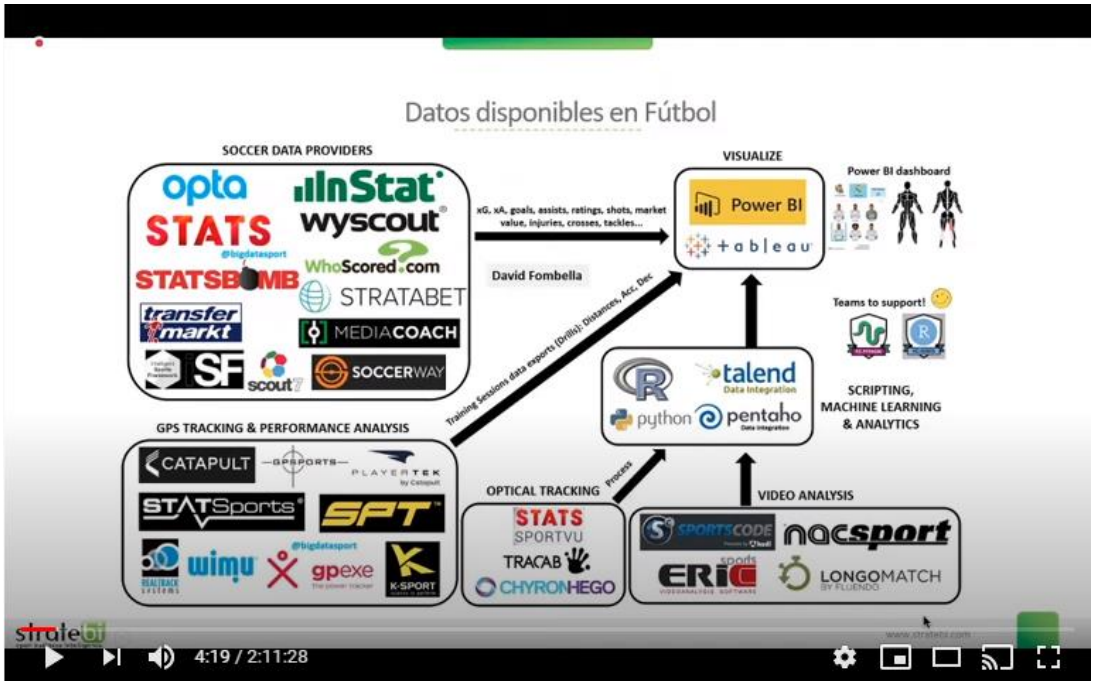

stratebi open business intelligence
Demo Finance
General
Localización
Usuarios
Balance
Cartera
Activos
Pasivos
Número de transacciones
781
Número de transaccionistas
545
Ganancias
1.856.550,13 €
Ganancias mensuales
38.678,13 €
Año
2014
2015
2016
2017
Mes
january
february
march
april
may
june
Categoría
Books
Cinema/TV
Electronics
Fashion
Home
Sports
Videogames
Localizaciones de las acciones
Search
Ireland
Kingdom
France
Spain
Portugal
Morocco
Transacciones
4
9
14
18
23
28
Ganancia %
70.89%
Gasto %
29.11%
Predicción y valor actual de las finanzas
Resultado previsto
Resultado actual
1.101.469,38 €
1.094.153,99 €
2016


Coste total
41.790.802,00 €
Personas por viaje
2,99
Viajes realizados
1.000
Media de viajes al mes
83,33
Gasto total de hombres y mujeres
HOMBRES (-40.9%)
33.606
MUJERES (-46.7%)
56.546
last 974 days
Pestañas
General
Destinos
Orígenes
Trayectos
Hoteles
Clientes
Año
Todas
Mes
Todas
Origen
Todas
Destino
Todas
Calidad del hotel
Todas
Edad
Todas
Cantidad de viajes y gasto realizado en ellos
Viajes
Coste total
Satisfacción media
85 %
Actividades realizadas
22,60%
Tour
29,40%
Museos
22,80%
Restauración
25,20%
Pub
Destinos más solicitados
Portugal
France
United States
Canada
Mexico
Sweden


PowerLinceBI
FÚTBOL
ANALYTICS
Top 10 jugadores
Mapa Jugadores
Mapa Formaciones y Demarcación
Formaciones Equipo
Estadísticas Equipo
Estadísticas Jugador
Ayuda
Creado por LinceBI


salesforce
Sales Cloud Dashboard
Country
Spain
United Kingdom
United States
State
Alabama
Extremadura
Nevada
Andalucía
Florida
North Carolina
Arkansas
Galicia
North Dakota
Asturias
Georgia
Ohio
Baleares
Illinois
País Vasco
Fiscal Year
2018
2019
Fiscal Quarter
Trim. 1
Trim. 2
Trim. 3
Trim. 4
Fiscal Month
enero
febrero
marzo
abril
mayo
junio
julio
agosto
septiembre
octubre
noviembre
diciembre
Stage Name
Calificación
Cerrada ganada
Cerrada perdida
Meeting Sche...
Negociación / ...
Propuesta / Pr...
Amount by Stage
Amount Trend
4.100,00 €
Opportunities
71
Signed Amount
283.322,00 €
Amount/Objective Last Quarter
101.477 €
Amount by State
Amount by Date and Fiscal Year
Trim. 1
Trim. 2
Trim. 3
Trim. 4


Tutorial de Introducción a Power BI
Ver más tarde
Compartir



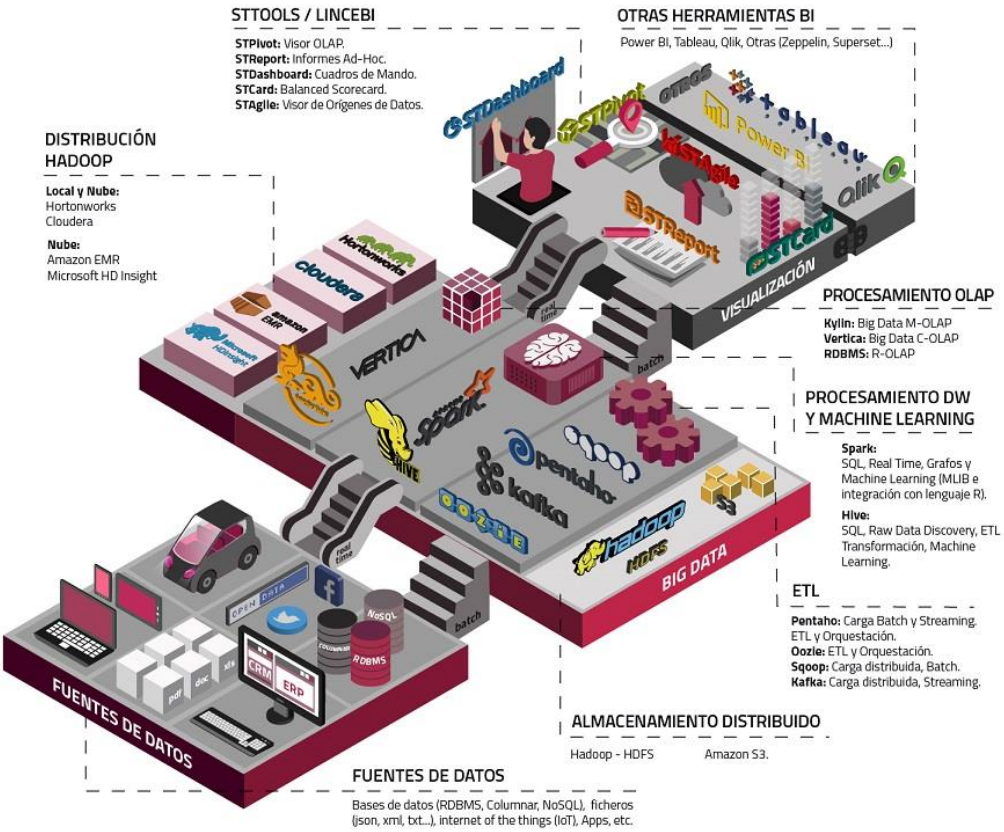
Sports Analytics con PowerBI

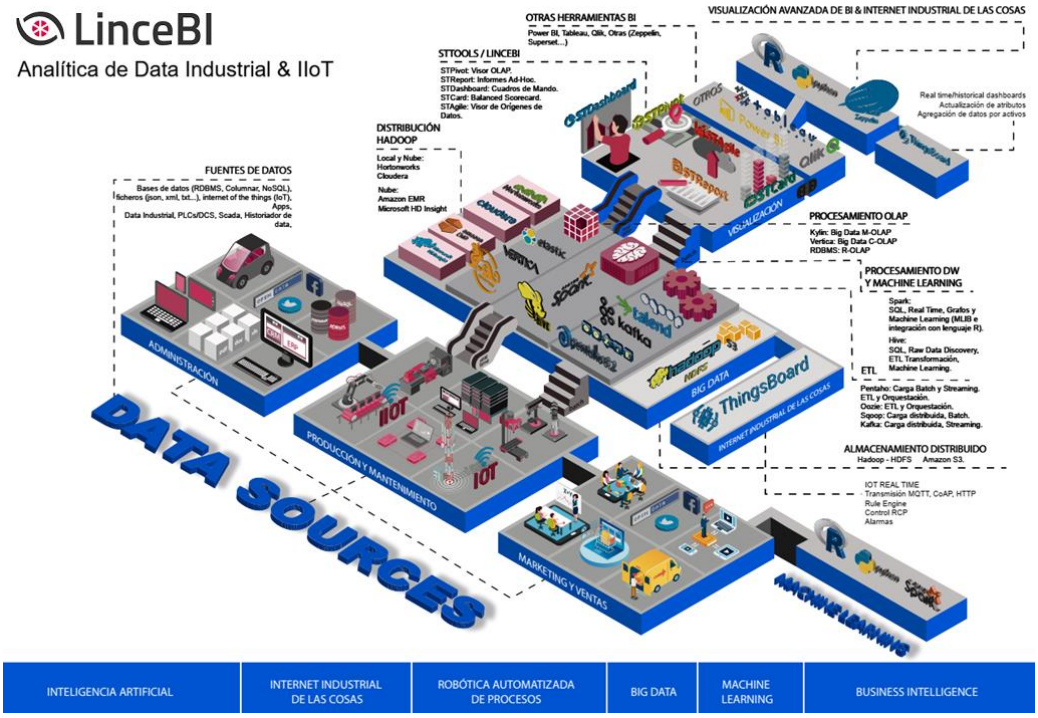
Recursos imprescindibles sobre PowerBI:

1. Integracion SAP - PowerBI
2. Futbol Analytics, lo que hay que saber
3. Dashboard de medicion de la calidad del aire en Madrid
4. Como funciona Microsoft Power BI? Videoturial de Introducción
5. Big Data para PowerBI
6. Como integrar Salesforce y PowerBI
7. Videotutorial: Usando R para Machine Learning con PowerBI
8. Las 50 claves para aprender y conocer PowerBI
9. PowerBI: Arquitectura End to End
10. Usando Python con PowerBI
11. PowerBI + Open Source = Sports Analytics
12. Comparativa de herramientas Business Intelligence
13. Use Case Big Data "Dashboards with Hadoop and Power BI"
14. Todas las presentaciones del Workshop 'El Business Intelligence del Futuro'
15. Descarga Paper gratuito: Zero to beautiful (Data visualization)

14. TECNOLOGÍAS

Recientemente, hemos sido nombrados Partners Certificados de Vertica, Talend, Microsoft, Snowflake, Kylligence, Pentaho, etc.


STTOOLS / LINCEBI
STPivot: Visor OLAP.
STReport: Informes Ad-Hoc.
STDashboard: Cuadros de Mando.
STCard: Balanced Scorecard.
STAgile: Visor de Orígenes de Datos.
OTRAS HERRAMIENTAS BI
Power BI, Tableau, Qlik, Otras (Zeppelin, Superset...)
DISTRIBUCIÓN HADOOP
Local y Nube:
Hortonworks
Cloudera
Nube:
Amazon EMR
Microsoft HD Insight
VISUALIZACIÓN
PROCESAMIENTO OLAP
Kylin: Big Data M-OLAP
Vertica: Big Data C-OLAP
RDBMS: R-OLAP
PROCESAMIENTO DW Y MACHINE LEARNING
Spark:
SQL, Real Time, Grafos y Machine Learning (MLIB e integración con lenguaje R).
Hive:
SQL, Raw Data Discovery, ETL Transformación, Machine Learning.
BIG DATA
ETL
Pentaho: Carga Batch y Streaming. ETL y Orquestación.
Oozie: ETL y Orquestación.
Sqoop: Carga distribuida, Batch.
Kafka: Carga distribuida, Streaming.
FUENTES DE DATOS
ALMACENAMIENTO DISTRIBUIDO
Hadoop - HDFS
Amazon S3.
FUENTES DE DATOS
Bases de datos (RDBMS, Columnar, NoSQL), ficheros (json, xml, txt...), internet of the things (IoT), Apps, etc.


LinceBI
Analítica de Data Industrial & IIoT
OTRAS HERRAMIENTAS BI
VISUALIZACIÓN AVANZADA DE BI & INTERNET INDUSTRIAL DE LAS COSAS
STTOOLS / LINCEBI
DISTRIBUCIÓN HADOOP
FUENTES DE DATOS
PROCESAMIENTO OLAP
PROCESAMIENTO DW Y MACHINE LEARNING
ETL
ALMACENAMIENTO DISTRIBUIDO
DATA SOURCES
ADMINISTRACIÓN
PRODUCCIÓN Y MANTENIMIENTO
MARKETING Y VENTAS
MACHINE LEARNING
INTELIGENCIA ARTIFICIAL
INTERNET INDUSTRIAL DE LAS COSAS
ROBÓTICA AUTOMATIZADA DE PROCESOS
BIG DATA
MACHINE LEARNING
BUSINESS INTELLIGENCE

15. INFORMACIÓN SOBRE STRATEBI

Stratebi es una empresa española, con sede en Madrid y oficinas en Barcelona, Alicante y Sevilla, creada por un grupo de profesionales con amplia experiencia en sistemas de información, soluciones tecnológicas y procesos relacionados con soluciones de Open Source y de inteligencia de Negocio.

Esta experiencia, adquirida durante la participación en proyectos estratégicos en compañías de reconocido prestigio a nivel internacional, se ha puesto a disposición de nuestros clientes.

Somos **Partners Certificados en Microsoft PowerBI** con una dilatada experiencia

Stratebi es la única empresa española que ha estado presente todos los Pentaho Developers celebrados en Europa habiendo organizado el de España.

En Stratebi nos planteamos como **objetivo** dotar a las compañías e instituciones, de herramientas escalables y adaptadas a sus necesidades, que conformen una estrategia Business Intelligence capaz de rentabilizar la información disponible. Para ello, nos basamos en el desarrollo de soluciones de Inteligencia de Negocio, mediante tecnología Open Source.

Stratebi **son** profesores y responsables de proyectos **del Master en Business Intelligence de la Universidad UOC, UCAM, EOI…**

Los profesionales de Stratebi son los creadores y autores del primer weblog en español sobre el mundo del Business Intelligence, Data Warehouse, CRM, Dashboards, Scorecard y Open Source. Todobi.com

Stratebi es partner de las principales soluciones Analytics: Microsoft Power BI, Talend, Pentaho, Vertica, Snowflake, Kyligence, Cloudera…

Todo Bi, se ha convertido en una referencia para el conocimiento y divulgación del Business Intelligence en español.

16. OTROS

Trabajamos en los principales sectores y con algunas de las compañías y organizaciones más importantes de España.

SECTOR PRIVADO

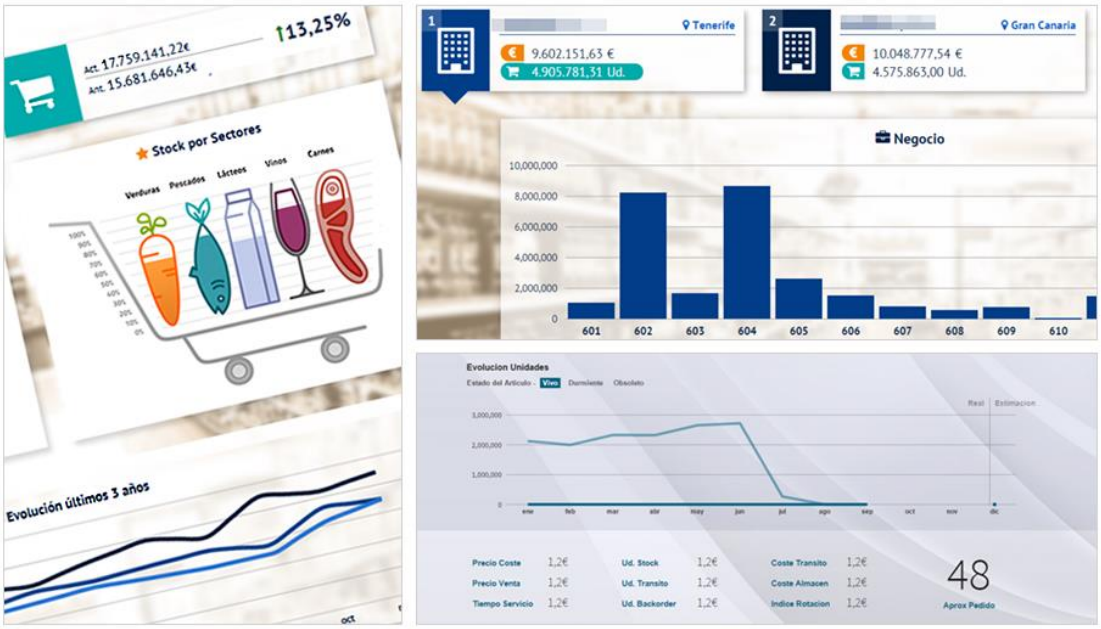
GLOBALIA
ITALFARMACO
pepecar.com
20:20mobile
EROSKI
amper
Unilever
educaweb (*)
EQUIFAX
BBVA
Blanco
PROSEGUR
FEMUR
Telefónica
FERMAX
BOSCH
AMNESTY INTERNATIONAL

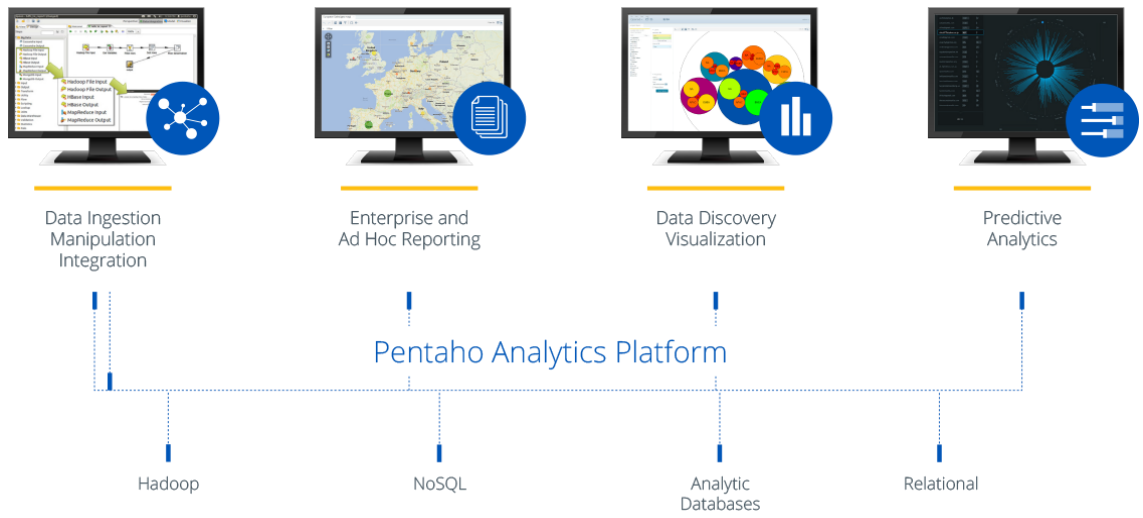
SECTOR PÚBLICO

Gobierno de Canarias
Consejería de Educación, Universidades y Sostenibilidad
UAM UNIVERSIDAD AUTONOMA DE MADRID
Caixa d'Enginyers
Diputació Tarragona
Ayuntamiento de Logroño
CSIC
Universitat de Lleida
IDI INSTITUT DE DIAGNÒSTIC PER LA IMATGE
Ayuntamiento de Arrecife
Gobierno de La Rioja
Dipsalut
Organisme Autònom de Salut Pública
Diputació de Girona
Universidad Zaragoza
AJUNTAMENT DE REUS
SaludMadrid
EOI Escuela de organización industrial

17. EJEMPLOS DE DESARROLLOS ANALYTICS

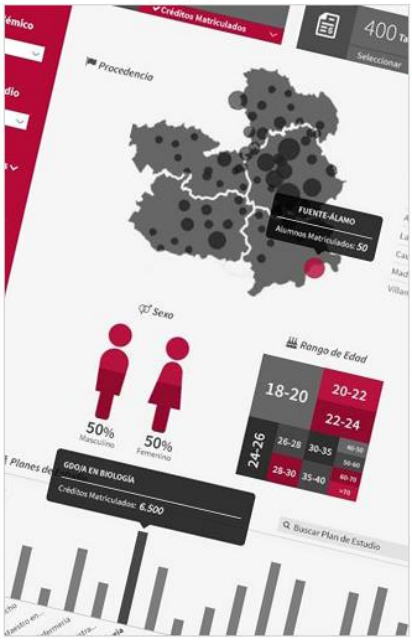
A continuación, se presentan **ejemplos de algunos screenshots** de cuadros de mando diseñados por Stratebi, con el fin de dar a conocer lo que se puede llegar a obtener, así como Demos Online en la web de Stratebi:



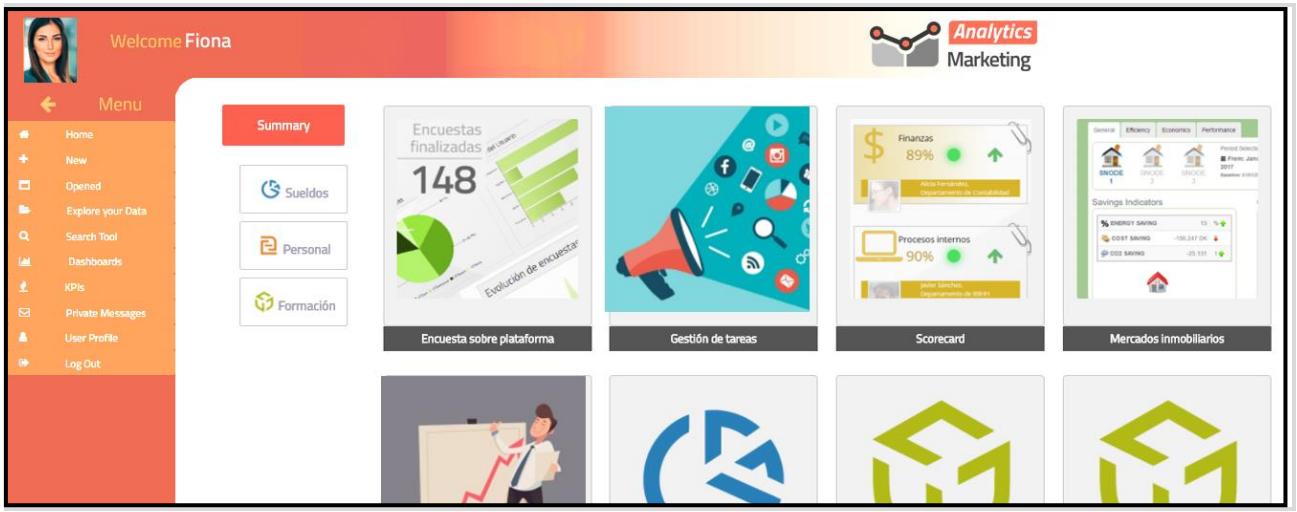



facebook DASHBOARD COMENTARIOS
Sexo
Sentimiento
Localización
Tipo de recurso
Evolución Comentarios por días


smart MADRID
Última Actualización
TOP 5 PUNTOS DE MEDICIÓN
PROXIMIDAD DE PUNTOS DE MEDIDA
EVOLUCIÓN INTENSIDAD DE TRÁFICO


Créditos Matriculados
Procedencia
FUENTE-ÁLAMO
Alumnos Matriculados: 50
Sexo
50%
50%
Rango de Edad
18-20
20-22
22-24
24-26
26-28
28-30
30-35
35-40
GDOJA EN BIOLOGÍA
Créditos Matriculados: 6.500
Buscar Plan de Estudio


181 Inscritos en Cursos
8 Desinscritos en Cursos
181 Inicios de Cursos
110 Inscritos en Eventos Virtuales
63 Inscritos en Debates
181 Inscritos en Talleres
USUARIOS ACTIVOS POR PAÍS
INSCRITOS EN CURSOS POR COMUNIDAD EDU
JORNADA COMPLETA
IDIOMAS
TIC
TIPO DE CENTRO
TIPO DE AUSENCIA
Nº de profesores por sexo
36%
64%
Nº total de profesores
1.587
Media de edad
43 años
Búsqueda


Welcome Fiona
Analytics
Marketing
Menu
Home
New
Opened
Explore your Data
Search Tool
Dashboards
KPIs
Private Messages
User Profile
Log Out
Summary
Sueldos
Personal
Formación
Encuestas finalizadas
148
Evolución de encuestas
Encuesta sobre plataforma
Gestión de tareas
Finanzas
89%
Procesos internos
90%
Scorecard
Savings Indicators
Mercados inmobiliarios