CATEGORIAS DE HERRAMIENTAS DE CIENCIAS DE DATOS

Bienvenido a las categorías de herramientas de ciencia de datos. Después de ver este vídeo, podrá enumerar las tareas que debe realizar un científico de datos, mostrar cómo la gestión de activos de código y la gestión de activos de datos ayudan a crear modelos, y describir cómo los entornos de ejecución y desarrollo implementan un modelo. Para que puedan resultar útiles, los datos sin procesar deben pasar por diversas categorías de tareas de la ciencia de datos, como la gestión de datos, la integración y transformación de datos, la visualización de datos, la creación de modelos, el despliegue de modelos y la supervisión y evaluación de modelos.

Para realizar estas tareas, necesita la administración de activos de datos, la administración de activos de código, los entornos de ejecución y los entornos de desarrollo. Veamos cómo cada categoría le permite aprovechar al máximo los datos sin procesar. La administración de datos es el proceso de recopilar, conservar y recuperar datos de forma segura, eficiente y rentable. Los datos se recopilan de muchas fuentes, como Twitter, Flipkart, Media, Sensors y más. Guarde los datos recopilados en un almacenamiento persistente para que estén disponibles siempre que los necesite. La integración y transformación de datos es el proceso de extracción, transformación y carga de datos. Esto se denomina «ETL». Algunos de estos datos se distribuyen en varios repositorios. Por ejemplo, una base de datos, un cubo de datos y archivos planos. Utilice el proceso de extracción para extraer datos de estos numerosos repositorios y guardarlos en un repositorio central, como un almacén de datos. Los almacenes de datos se utilizan principalmente para recopilar y almacenar grandes cantidades de datos para su análisis. A continuación, la transformación de datos es el proceso de transformar los valores, la estructura y el formato de los datos. Tras extraer los datos, el siguiente paso es transformarlos. En este ejemplo, los datos de altura y peso deben transformarse en métricas. Y una vez transformados los datos, es el momento de cargarlos. Los datos transformados se vuelven a cargar en el almacén de datos. La visualización de datos es la representación gráfica de los datos y la información. Puede utilizar la visualización para representar los datos en forma de gráficos, gráficos, mapas, animaciones, etc. Además, la visualización de datos transmite los datos de forma más eficaz a los responsables de la toma de decisiones. Es un paso crucial en el proceso de ciencia de datos. Las diversas formas de visualización de datos incluyen un gráfico de barras, que compara el tamaño de cada componente, un mapa de árbol, que muestra datos jerárquicos, un gráfico de líneas, que traza una serie de puntos de datos a lo largo del tiempo, y un gráfico de mapa, que muestra los datos por ubicación.

Los gráficos de mapas también se pueden aplicar a otras ubicaciones, como sitios web. Ahora, la creación de modelos es el siguiente paso. Aquí es donde se entrenan los datos y se analizan los patrones con algoritmos de aprendizaje automático. El sistema «aprende» a proporcionar predicciones o decisiones por sí mismo. Luego, puede usar este modelo para hacer predicciones sobre datos nuevos e invisibles. La creación de modelos se puede realizar mediante un servicio llamado IBM Watson Machine Learning. Proporciona una gama completa de herramientas y servicios para crear modelos. El siguiente paso es la implementación del modelo: el proceso de integración de un modelo desarrollado en un entorno de producción. En el despliegue de modelos, se pone a disposición de las aplicaciones de terceros un modelo de aprendizaje automático mediante API.

Los usuarios empresariales pueden acceder a los datos e interactuar con ellos a través de estas aplicaciones de terceros. Y esto les ayuda a tomar decisiones basadas en datos. Por ejemplo, los servicios de colaboración e implementación de SPSS se pueden utilizar para implementar cualquier tipo de activo creado por el paquete de herramientas de software de SPSS. La supervisión y la evaluación de los modelos realizan controles de calidad continuos para garantizar la precisión, imparcialidad y solidez del modelo.

La supervisión de modelos utiliza herramientas como Fiddler para realizar un seguimiento del rendimiento de los modelos implementados en un entorno de producción. Ahora, la evaluación del modelo utiliza métricas de evaluación como la puntuación F1, la tasa de positivos reales o la suma de los errores cuadrados para comprender el rendimiento de un modelo. Un ejemplo bien conocido es la báscula IBM Watson Open, que monitorea continuamente los modelos implementados de aprendizaje automático y aprendizaje profundo. Mejorará la precisión y la calidad de sus predicciones. Ahora que hemos revisado las categorías de tareas de ciencia de datos, veamos algunas de las herramientas que las respaldan.

La gestión de activos de código proporciona una vista unificada en la que se gestiona un inventario de activos. Cuando desee desarrollar un modelo, es posible que necesite actualizarlo, corregir errores o mejorar las características del código de forma gradual. Todo esto requiere el control de versiones. Los desarrolladores utilizan el control de versiones para realizar un seguimiento y gestionar los cambios en el código de un proyecto de software. Cuando trabajan en un modelo, los equipos utilizan un repositorio centralizado donde todos pueden cargar, editar y gestionar los archivos de código simultáneamente. La colaboración permite que distintas personas compartan y actualicen el mismo proyecto de forma conjunta. GitHub es un buen ejemplo de plataforma de administración de activos de código. Está basada en la web y proporciona funciones para compartir, colaborar y controlar el acceso. Como científico de datos, desea almacenar y organizar adecuadamente todas sus imágenes, vídeos , texto y otros datos en una ubicación central. También quiere controlar quién puede acceder a sus datos, editarlos y gestionarlos. La gestión de activos de datos, también denominada gestión de activos digitales (DAM), consiste en organizar y gestionar datos importantes recopilados de diferentes fuentes. La DAM se realiza en una plataforma DAM que permite el control de versiones y la colaboración. Las plataformas DAM también admiten la replicación, el respaldo y la administración de los derechos de acceso de los datos almacenados. Los entornos de desarrollo, también denominados entornos de desarrollo integrados o «IDE», proporcionan un espacio de trabajo y herramientas para desarrollar, implementar, ejecutar, probar e implementar el código fuente. Los IDE como IBM Watson Studio proporcionan herramientas de prueba y simulación para emular el mundo real, de modo que pueda ver cómo se comportará su código una vez desplegado. Un entorno de ejecución tiene bibliotecas para compilar el código fuente y los recursos del sistema que ejecutan y verifican el código. Los entornos de ejecución basados en la nube no están vinculados a ningún hardware o software específico y ofrecen herramientas como IBM Watson Studio para el preprocesamiento de datos, el entrenamiento de modelos y el despliegue. Por último, las herramientas visuales totalmente integradas, como IBM Watson Studio e IBM Cognos Dashboard Embedded, cubren todos los componentes de herramientas anteriores y se pueden utilizar para desarrollar modelos de aprendizaje profundo y aprendizaje automático. En este vídeo, aprendió que las categorías de tareas de la ciencia de datos son: 100de datos, integración y transformación de datos, visualización de datos, creación de modelos , despliegue de modelos y supervisión y evaluación de modelos. Las tareas de ciencia de datos están respaldadas por la administración de activos de datos, la administración de activos de código , los entornos de ejecución y los entornos de desarrollo.

HERRAMIENTAS DE CÓDIGO ABIERTO PARA LA CIENCIA DE DATOS, PARTE I

Bienvenido a «Herramientas de código abierto para la ciencia de datos, parte 1». Después de ver este vídeo, podrá: Enumerar las herramientas de gestión de datos de código abierto. Enumere las herramientas de integración y transformación de datos de código abierto. Enumere las herramientas de visualización de datos. Enumere las herramientas modelo para crear, implementar, monitorear y evaluar, y enumere las herramientas para la administración de activos de código y datos.

Por lo tanto, las herramientas de administración de datos de código abierto más utilizadas son las bases de datos relacionales, como MySQL y PostgreSQL. Además, hay bases de datos NoSQL como MongoDB, Apache CouchDB y Apache Cassandra. Además, existen herramientas basadas en archivos como el sistema de archivos Hadoop o sistemas de archivos en la nube como Ceph. También dispone de una herramienta de búsqueda elástica que almacena datos de texto, incluida la creación de un índice de búsqueda para recuperar rápidamente los documentos.

Ahora, la tarea de integración y transformación de datos en el mundo clásico del almacenamiento de datos consiste en extraer, transformar y cargar (ETL). Los científicos de datos suelen proponer extraer, cargar y transformar (ELT), ya que los datos se descargan en algún lugar y el ingeniero o científico de datos se encarga de la transformación de los datos.

Surgió otro término para este proceso: refinería y limpieza de datos. Las herramientas de integración y transformación de datos de código abierto más utilizadas son las siguientes: Apache AirFlow, que fue creada originalmente por Airbnb. KubeFlow, que permite ejecutar procesos de ciencia de datos sobre Kubernetes. Apache Kafka, que se originó en LinkedIn. Apache Nifi, que ofrece un editor visual muy agradable. Apache SparkSQL le permite usar ANSI SQL y se amplía para calcular clústeres de miles de nodos, y NodeRed también incluye un editor visual. Además, el consumo de recursos de NodeRed es tan bajo que incluso se ejecuta en dispositivos pequeños como una Raspberry Pi. Ahora analicemos las herramientas de visualización de datos de código abierto más utilizadas. Debe distinguir entre las bibliotecas de programación en las que debe usar código o herramientas que contengan una interfaz de usuario. Pixie Dust también es una biblioteca, pero tiene una interfaz de usuario que facilita el trazado en Python. Un enfoque similar utiliza Hue, que puede crear visualizaciones a partir de consultas SQL. Mientras que Kibana, una aplicación web de exploración y visualización de datos, se limita a Elasticsearch (proveedor de datos). Y, por último, Apache Superset es una aplicación web de exploración y visualización de datos. La implementación del modelo es un paso crucial. Una vez que haya creado un modelo de aprendizaje automático capaz de predecir algunos aspectos críticos del futuro, debería hacerlo utilizable por otros desarrolladores y convertirlo en una API. Actualmente, Apache PredictionIO solo es compatible con los modelos Apache Spark ML para su implementación, pero la compatibilidad con todas las bibliotecas está prevista. Seldon es un producto interesante, ya que es compatible con casi todos los marcos, incluidos TensorFlow, Apache SparkML, R y scikit learn.

Curiosamente, puede ejecutarse sobre Kubernetes y Redhat OpenShift. Otra forma de implementar los modelos de SparkML es mLeap.

Por último, TensorFlow puede servir cualquier modelo de flujo tensorial mediante el servicio TensorFlow. Puede ser un dispositivo integrado, como una Raspberry Pi o un teléfono inteligente con TensorFlow lite, e implementarse en un navegador web con TensorFlow punto JS. La supervisión del modelo también es un paso importante. Una vez que haya implementado un modelo de aprendizaje automático, querrá realizar un seguimiento de su rendimiento de predicción mientras llegan nuevos datos para mantener los modelos obsoletos. Algunos ejemplos son los siguientes: ModelDB es una base de metadatos de modelos de máquinas en la que se almacena y consulta información sobre los modelos. Es compatible de forma nativa con Apache Spark ML Pipelines y scikit-learn. También se utiliza ampliamente una herramienta genérica y multipropósito llamada Prometheus. Aunque no está hecha específicamente para el monitoreo de modelos de aprendizaje automático, se usa para este propósito. El rendimiento del modelo se mide por algo más que por su precisión. El sesgo del modelo en contra de los grupos protegidos, como el género o la raza, también es importante. El kit de herramientas de código abierto IBM AI Fairness 360 detecta y mitiga los sesgos en los modelos de aprendizaje automático. Estos modelos, especialmente los modelos de aprendizaje profundo basados en redes neuronales, pueden ser objeto de ataques adversos en los que un atacante intenta engañar al modelo con datos manipulados o controlándolos. La caja de herramientas Adversarial Robustness 360 de IBM detecta la vulnerabilidad frente a los ataques adversos y aprovecha el modelo para ser más robusto.

Por último, los modos de aprendizaje automático suelen considerarse como una caja negra en la que se aplica un poco de magia. El kit de herramientas IBM AI Explainability 360 aborda ese problema al encontrar ejemplos similares en un conjunto de datos para presentarlos al usuario final para su comparación manual. El kit de herramientas IBM AI Explainability 360 también puede abordar el entrenamiento de un modelo de aprendizaje automático más simple para explicar la responsabilidad de las diferentes variables de entrada en la decisión final del modelo.

Por lo tanto, la elección de las herramientas de gestión de activos de código se ha vuelto bastante sencilla. Git es ahora el estándar de facto para la gestión de activos de código, también conocida como gestión de versiones o control de versiones. En torno a Git surgieron varios servicios. El más destacado es GitHub, pero el segundo lugar es GitLab, con la ventaja de que la plataforma es totalmente de código abierto y se puede alojar y gestionar por tu cuenta. Otra opción es Bitbucket.

La gestión de activos de datos, también conocida como gobierno de datos o linaje de datos, es una parte crucial de la ciencia de datos de nivel empresarial. Los datos tienen que estar versionados y anotados con metadatos. Apache Atlas es una herramienta de este tipo que apoya esta tarea. Otro proyecto interesante es ODPi Egeria, gestionado a través de la Fundación Linux, que es un ecosistema abierto que ofrece un conjunto de API , tipos y protocolos de intercambio abiertos que los repositorios de metadatos utilizan para compartir e intercambiar datos. Y, por último, Kylo es una plataforma de software de gestión de datos de código abierto, con un amplio soporte para las tareas de gestión de activos de datos. En este vídeo, descubrirá lo siguiente: las herramientas de gestión de datos son MySQL, PostgreSQL, MongoDB, Apache CouchDB, Apache Cassandra, Hadoop File System, Ceph y elastic search. Las herramientas de integración y transformación de datos son Apache AirFlow, KubeFlow, Apache Kafka, Apache Nifi, Apache SparkSQL y NodeRed. Las herramientas de visualización de datos son Pixie Dust, Hue, Kibana y Apache Superset. Las herramientas de implementación de modelos son Apache PredictionIO, Seldon, Kubernetes, Redhat OpenShift, Mleap, el servicio TensorFlow, TensorFlow lite y TensorFlow punto JS. Las herramientas de supervisión de modelos son ModelDB, Prometheus, IBM AI Fairness 360, IBM Adversarial Robustness 360 Toolbox e IBM AI Explainability 360. Las herramientas de gestión de activos de código son Git, GitHub, GitLab y Bitbucket. Por último, las herramientas de gestión de activos de datos son Apache Atlas, ODPi Egeria y Kylo.

* **BM Watson Studio:** Diseñado como un entorno integrado, Watson Studio simplifica el desarrollo, la formación y el despliegue de modelos. Ofrece compatibilidad con varios lenguajes y marcos de trabajo, como Python, R y TensorFlow, además de funciones de colaboración, herramientas de preparación de datos y opciones de despliegue versátiles.
* IBM AutoAI**:** IBM AutoAI, una función notable integrada en Watson Studio, agiliza el proceso de construcción de modelos de aprendizaje automático. Mediante la exploración dinámica de varios algoritmos e hiperparámetros, pretende identificar el modelo óptimo para un conjunto de datos determinado.
* **IBM Watson OpenScale:** Como plataforma para supervisar y gestionar modelos de IA en producción, Watson OpenScale desempeña un papel fundamental a la hora de garantizar la equidad, la explicabilidad y la mitigación de sesgos de los modelos. Proporciona información sobre el rendimiento del modelo y su evolución a lo largo del tiempo, lo que facilita la toma de decisiones informadas.
* **IBM Watson Aprendizaje automático:** Watson Aprendizaje automático, disponible como servicio en la plataforma IBM Cloud, permite a los usuarios escalar su formación y despliegue de modelos de aprendizaje automático. Da soporte sin problemas a marcos populares como TensorFlow, PyTorch y Scikit-learn, y ofrece API para una integración perfecta con otras aplicaciones.

**Herramientas de código abierto. Parte II**

Bienvenido a Herramientas de código abierto para la ciencia de datos, parte 2. Después de ver este vídeo, podrá comparar y contrastar diferentes herramientas de código abierto y describir las características relevantes de las herramientas de código abierto.

Actualmente, el entorno de desarrollo más famoso que utilizan los científicos de datos es «Jupyter», que surgió como una herramienta para la programación interactiva en Python. Jupyter ahora es compatible con más de cien hids de programación diferentes a través de «núcleos». Esto encapsula el entorno de ejecución de los diferentes lenguajes de programación. Una propiedad clave de Jupyter Notebooks es unificar la documentación, el código, los resultados del código, los comandos de shell y las visualizaciones en un solo documento.

Jupyter lab es la próxima versión de Jupyter Notebooks y, a largo plazo, sustituirá a Jupyter Notebooks. La abundancia de cambios arquitectónicos hace que Jupyter sea más moderno y modular. Desde el punto de vista del usuario, la principal diferencia entre Jupyter Lab y Jupyter Notebooks es la posibilidad de abrir distintos tipos de archivos, incluidos los cuadernos de Jupyter, datos y terminales, y luego organizarlos en el lienzo. Aunque se ha vuelto a implementar desde cero, Apache Zeppelin se inspiró en los cuadernos de Jupyter y ofrece una experiencia similar. Un elemento diferenciador clave es la capacidad de representación gráfica integrada.

En Jupyter Notebooks, es necesario utilizar bibliotecas externas en Zeppelin y el trazado no requiere codificación. También puede ampliar las capacidades mediante el uso de bibliotecas adicionales.

RStudio es uno de los entornos de desarrollo más antiguos para la estadística y la ciencia de datos. RStudio tiene sus orígenes en el año 2011. Ejecuta exclusivamente R y todas las bibliotecas R asociadas. En el entorno R, es posible desarrollar Python. R está perfectamente integrado en la herramienta Jupyter y proporciona una experiencia de usuario óptima. RStudio unifica la programación, la ejecución, la depuración, el acceso remoto a los datos, la exploración de datos y la visualización en una sola herramienta.

Por último, Spyder intenta imitar el comportamiento de RStudio para llevar su funcionalidad al mundo de Python. Aunque no está a la altura de la funcionalidad de RStudio, los científicos de datos lo consideran una alternativa. En el mundo de Python, Jupyter se usa más. Este diagrama muestra que Spyder integra el código, la documentación y las visualizaciones, entre otros, en un solo lienzo. A veces, los datos no caben en la capacidad de almacenamiento o memoria principal de un único ordenador. Por lo tanto, existen entornos de ejecución en clústeres.

El famoso Apache Spark es uno de los proyectos de Apache más activos que se utilizan en todos los sectores, incluidas muchas empresas de la lista Fortune 500. La propiedad clave de Apache Spark es la escalabilidad lineal. Esto significa que si duplica el número de servidores de un clúster, prácticamente duplicará su rendimiento.

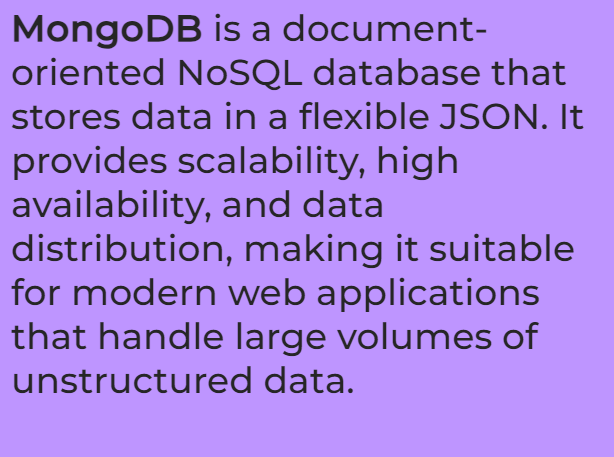
Apache Flink se desarrolló después de que Apache Spark siguiera ganando cuota de mercado. La diferencia clave entre Apache Spark y Apache Flink es que Apache Spark es un motor de procesamiento de datos por lotes, capaz de procesar grandes cantidades de datos uno por uno o archivo por archivo. Mientras que Apache Flink es una imagen de procesamiento de flujos cuyo objetivo principal es procesar flujos de datos en tiempo real. Si bien ambos motores admiten ambos paradigmas de procesamiento de datos, Apache Spark es la opción ideal para la mayoría de los casos de uso.

Después de Apache Spark y Apache Flink, Ray es uno de los últimos avances en los entornos de ejecución de la ciencia de datos y tiene un claro enfoque en el entrenamiento de modelos de aprendizaje profundo a gran escala. Veamos las herramientas de código abierto para científicos de datos, que son visuales y totalmente integradas. Esto significa que no es necesario tener conocimientos de programación. Las herramientas admiten un subconjunto de tareas importantes que incluyen la integración y transformación de datos, la visualización de datos y la creación de modelos.

KNIME se originó en la Universidad de Constanza en 2004. Como puede ver, KNIME tiene una interfaz de usuario visual con funciones de arrastrar y soltar. Tiene capacidades de visualización integradas. Además, se puede ampliar programando en R y Python e incluso tiene conectores a Apache Spark. Orange es otro representante de este grupo de herramientas. Es menos flexible que KNIME, pero es más fácil de usar. En este vídeo, conocerá las tareas más comunes de la ciencia de datos y qué herramientas de código abierto son relevantes.

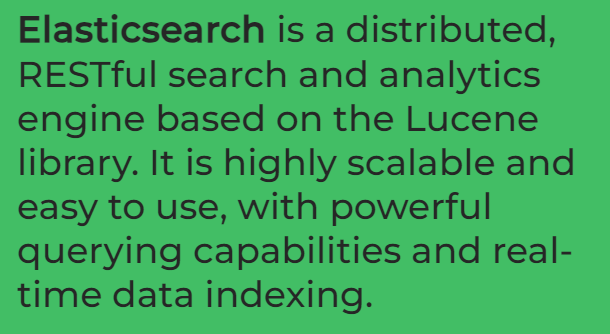
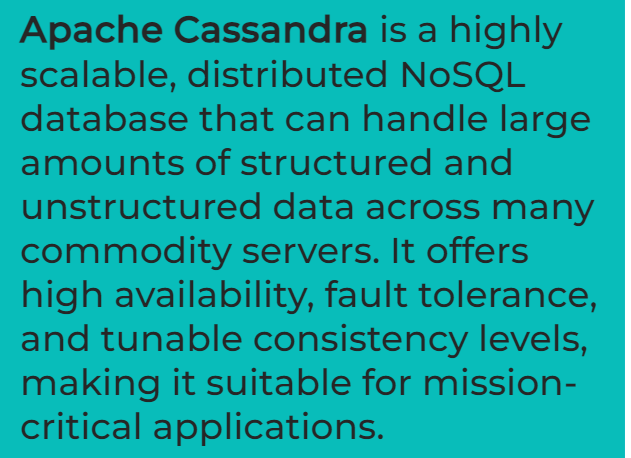
Texto

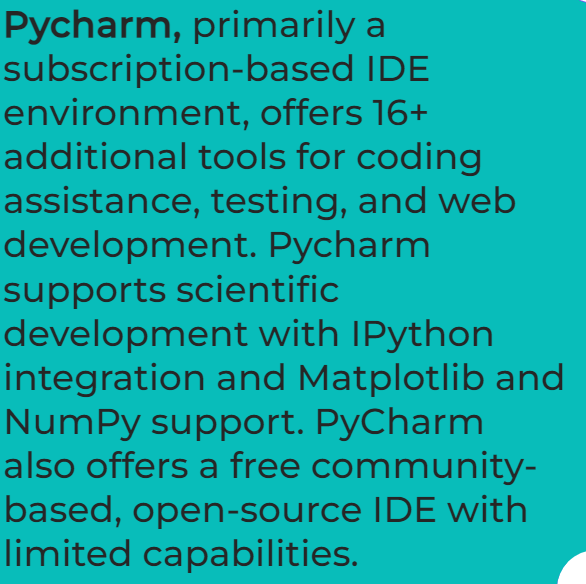
Descripción generada automáticamenteTexto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto, Chat o mensaje de texto

Descripción generada automáticamenteTexto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamenteTexto, Chat o mensaje de texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto

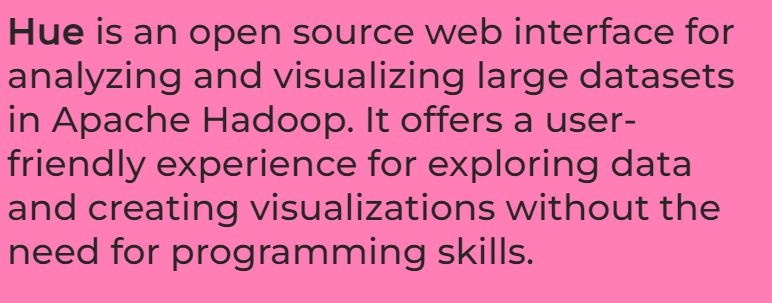
Descripción generada automáticamenteTexto

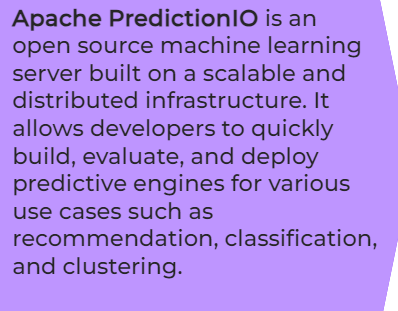
Descripción generada automáticamenteTexto, Chat o mensaje de texto

Descripción generada automáticamenteTexto, Chat o mensaje de texto

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamenteTexto, Aplicación, Chat o mensaje de texto

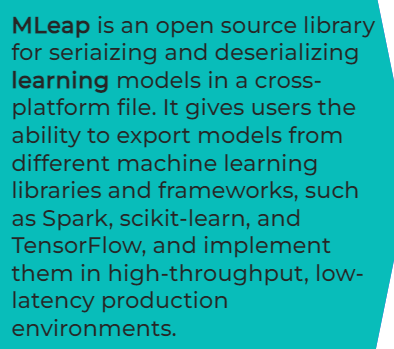
Descripción generada automáticamenteTexto, Aplicación, Chat o mensaje de texto

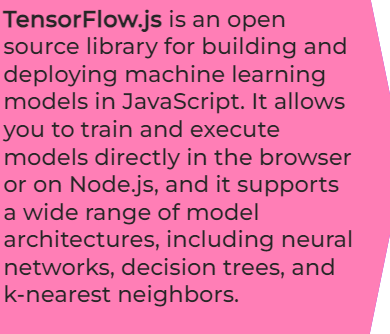
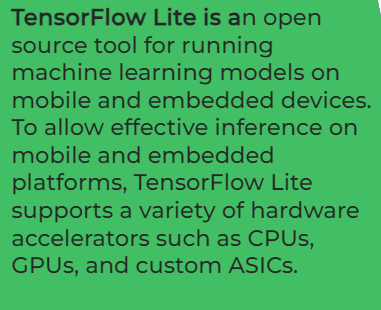
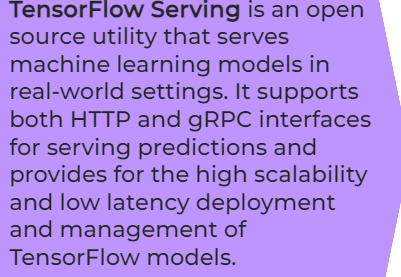
Descripción generada automáticamenteTexto

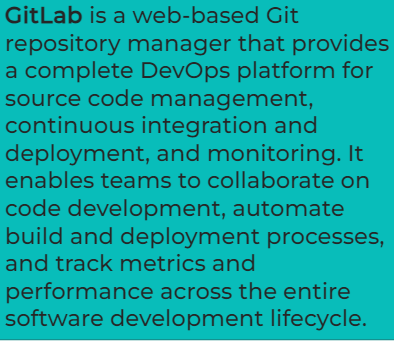
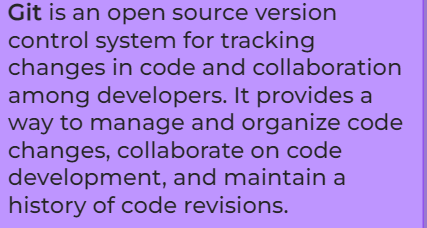
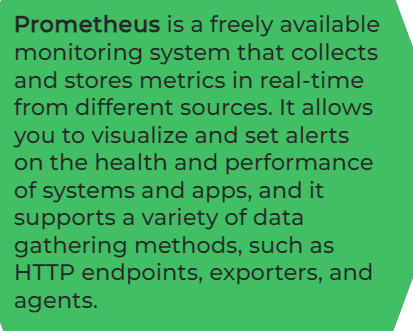
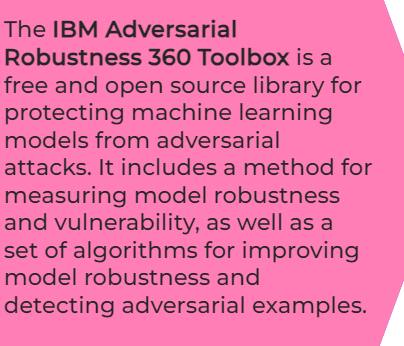
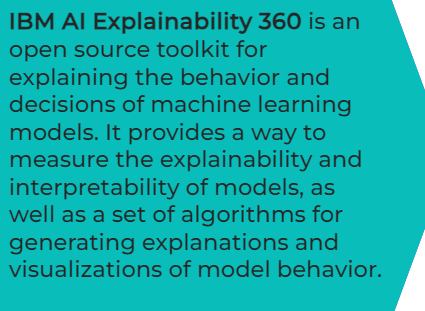
Descripción generada automáticamenteTexto

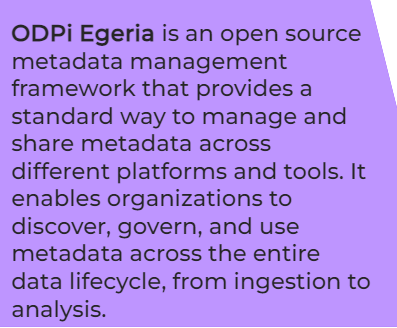
Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamenteTexto, Chat o mensaje de texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

**Herramientas comerciales para ciencia de datos**

Anteriormente cubrimos las herramientas de código abierto para la ciencia de datos. Ahora, veamos las opciones comerciales que encontrará en muchos proyectos empresariales. Revisemos nuestra descripción general de las diferentes categorías de herramientas. En la administración de datos, la mayoría de los datos relevantes de una empresa se almacenan en Oracle Database, Microsoft SQL Server o IBM Db2. Aunque las bases de datos de código abierto están ganando popularidad, estos tres productos de gestión de datos siguen siendo considerados como el estándar de la industria. No desaparecerán en un futuro cercano. No se trata solo de funcionalidad. Los datos están en el centro de todas las organizaciones, y la disponibilidad de soportes comerciales desempeña un papel importante. Los soportes comerciales se entregan directamente de proveedores de software, socios influyentes y redes de soporte. Cuando nos centramos en las herramientas de integración de datos comerciales, estamos hablando de «extraer, transformar y cargar» o «ETL» herramientas. Según un Cuadrante Mágico de Gartner, Informatica Powercenter e IBM InfoSphere DataStage son los líderes, seguidos por los productos de SAP, Oracle, SAS, Talend y Microsoft. Estas herramientas admiten el diseño y la implementación de canalizaciones de procesamiento de datos ETL a través de una interfaz gráfica. También proporcionan conectores a la mayoría de los sistemas de información de destino comerciales y de código abierto. Por último, Watson Studio Desktop incluye un componente denominado Refinería de datos, que permite definir y ejecutar procesos de integración de datos en un estilo de hoja de cálculo. En el entorno comercial, las visualizaciones de datos utilizan herramientas de inteligencia empresarial o «BI». Su objetivo principal es crear informes y paneles en vivo visualmente atractivos y fáciles de entender. Los ejemplos comerciales más destacados son: Tableau, Microsoft Power BI e IBM Cognos Analytics. Otro tipo de visualización se dirige a los científicos de datos en lugar de a los usuarios habituales. Un problema de ejemplo podría ser «¿Cómo se pueden relacionar las diferentes columnas de una tabla entre sí?» Este tipo de funcionalidad está contenida en Watson Studio Desktop. Si desea crear un modelo de aprendizaje automático utilizando una herramienta comercial, debe considerar el uso de un producto de minería de datos. Los más destacados de estos tipos de productos son: SPSS Modeler y SAS Enterprise Miner. Además, una versión de SPSS Modeler también está disponible en Watson Studio Desktop, basada en la versión en la nube de la herramienta. Hablaremos más sobre las herramientas basadas en la nube en el siguiente vídeo. En el software comercial, la implementación de modelos está estrechamente integrada en el proceso de construcción de modelos. Este diagrama muestra un ejemplo de los Servicios de colaboración e implementación de SPSS que se utilizan para implementar cualquier tipo de activo creado por el conjunto de herramientas de software SPSS. Otros proveedores utilizan el mismo tipo de proceso. El software comercial también puede exportar modelos en formato abierto. Por ejemplo, SPSS Modeler admite la exportación de modelos como Predictive Model Markup Language, o PMML, que pueden ser leídos por muchos otros paquetes de software comerciales y abiertos. El monitoreo de modelos es una nueva disciplina y actualmente no hay herramientas comerciales relevantes disponibles. Como resultado, el código abierto es la primera opción. Lo mismo ocurre con la administración de activos de código. El código abierto con Git y GitHub es el estándar efectivo. La gestión de activos de datos, a menudo denominada gobernanza de datos o linaje de datos, es una parte crucial de la ciencia de datos de nivel empresarial. Los datos deben ser versionados y anotados mediante metadatos. Los proveedores, incluidos Informatica Enterprise Data Governance e IBM, proporcionan herramientas para estas tareas específicas. IBM InfoSphere Information Governance Catalog cubre funciones como diccionario de datos, lo que facilita el descubrimiento de activos de datos. Cada activo de datos se asigna a un administrador de datos: el propietario de los datos. El propietario de los datos es responsable de ese activo de datos y puede ser contactado. El linaje de datos también está cubierto; esto permite al usuario realizar un seguimiento a través de los pasos de transformación seguidos en la creación de los activos de datos. El linaje de datos también incluye una referencia a los datos de origen reales. Se pueden agregar reglas y políticas para reflejar los complejos requisitos regulatorios y empresariales en materia de privacidad y retención de datos. Watson Studio es un entorno de desarrollo totalmente integrado para científicos de datos. Por lo general, se consume a través de la nube, y cubriremos más sobre él en una lección posterior. También hay una versión de escritorio disponible. Watson Studio Desktop combina portátiles Jupyter con herramientas gráficas para maximizar el rendimiento de los científicos de datos. Watson Studio, junto con Watson Open Scale, es una herramienta totalmente integrada que cubre todo el ciclo de vida de la ciencia de datos y todas las tareas que hemos discutido anteriormente. Hablaremos más de ambos en la próxima lección. pero solo tenga en cuenta que pueden implementarse en un centro de datos local encima de Kubernetes o RedHat OpenShift. Otro ejemplo de una herramienta comercial totalmente integrada es H2O Driverless AI, que cubre todo el ciclo de vida de la ciencia de datos. En esta lección, ha aprendido cómo las herramientas comerciales admiten las tareas más comunes de ciencia de datos. En el siguiente vídeo, descubriremos herramientas de ciencia de datos que están disponibles exclusivamente en la nube.

[**Herramientas basadas en la nube para la ciencia de datos**](https://www.coursera.org/learn/open-source-tools-for-data-science/lecture/MYYx2/cloud-based-tools-for-data-science)

**Bienvenido a «Herramientas basadas en la nube para la ciencia de datos». Tras ver este vídeo, podrá: describir cómo las herramientas comerciales en la nube facilitan las tareas de ciencia de datos y explicar cómo la integración permite utilizar las mismas herramientas para varias tareas. Veamos nuevamente la descripción general de las diferentes categorías de herramientas. Dado que los productos en la nube son una especie más nueva, siguen la tendencia de tener varias tareas integradas en las herramientas. Esta integración es aplicable a las tareas marcadas en verde en el diagrama. Empecemos con la categoría de herramientas visuales totalmente integradas. Como estas herramientas introducen un componente que permite ejecutar a gran escala los flujos de trabajo de la ciencia de datos en clústeres de cómputo, hemos cambiado el título y hemos añadido la palabra «Plataforma». Estos clústeres se componen de varios servidores , de forma transparente para el usuario y en segundo plano. Watson Studio y Watson OpenScale cubren el ciclo de vida completo del desarrollo de todas las tareas de ciencia de datos, aprendizaje automático e inteligencia artificial (IA). Otro ejemplo es el aprendizaje automático de Microsoft Azure. También es una oferta totalmente alojada en la nube que soporta todo el ciclo de vida de desarrollo de todas las tareas de ciencia de datos, aprendizaje automático e inteligencia artificial. Por último, otro ejemplo es H2O Driverless AI. Aunque es un producto que se descarga e instala, existe un despliegue con un solo clic para los proveedores de servicios en la nube estándar. Dado que el proveedor de la nube no se encarga de las operaciones ni el mantenimiento, como ocurre con Watson Studio, Open Scale y Azure Machine Learning, este modelo de entrega debe ser distinto del de plataforma o software como servicio (PaaS o SaaS). En la gestión de datos, con algunas excepciones, existen versiones de software como servicio (SaaS) de las herramientas comerciales y de código abierto existentes. El proveedor de la nube opera la herramienta por usted en la nube. Por ejemplo, el proveedor de la nube opera el producto realizando una copia de seguridad de los datos y configurando e instalando las actualizaciones. Algunas herramientas patentadas solo están disponibles en un único proveedor de servicios en la nube. Un ejemplo de este tipo de servicio es Amazon Web Services DynamoDB, que es una base de datos NoSQL. Permite almacenar y recuperar datos en un formato clave-valor o de almacén de documentos. La estructura de datos del documento más destacada es JSON. Otro tipo de servicio de este tipo es Cloudant, que es una oferta de base de datos como servicio. Pero, en segundo plano, se basa en el código abierto Apache CouchDB. La ventaja es que las tareas operativas complejas, como la actualización, la copia de seguridad, la restauración y el escalado, las realiza el proveedor de la nube. Sin embargo, la oferta de servicios de Cloudant es compatible con CouchDB. Por lo tanto, la aplicación migra a otro servidor de CouchDB sin realizar ningún cambio en la aplicación. IBM también ofrece Db2 como servicio. Es un ejemplo de una base de datos comercial disponible como oferta SaaS en la nube, lo que elimina las tareas operativas del usuario. Ahora analicemos las herramientas de integración de datos comerciales, que incluyen herramientas de extracción, transformación y carga (ETL) y herramientas de extracción, carga y transformación (ELT). Esto significa que los pasos de transformación no los lleva a cabo un equipo de integración de datos, sino que los lleva a cabo el científico o el ingeniero de datos. Dos herramientas comerciales de integración de datos muy utilizadas son Informatica Cloud Data Integration y Data Refinery de IBM. Data Refinery forma parte de IBM Watson Studio. Permite transformar grandes cantidades de datos sin procesar en información consumible de calidad en una interfaz de usuario similar a una hoja de cálculo. Por lo tanto, el mercado de herramientas de visualización de datos en la nube es enorme y todos los principales proveedores de servicios en la nube tienen una. Un ejemplo de una empresa más pequeña que ofrece una herramienta de visualización de datos basada en la nube es Datameer. IBM ofrece su famosa suite de inteligencia empresarial Cognos como solución en la nube. IBM Data Refinery también ofrece funciones de exploración y visualización de datos en Watson Studio. Una vez más, estos son ejemplos de un ecosistema comercial que cambia y crece rápidamente entre muchos proveedores establecidos y emergentes. En Watson Studio, varias visualizaciones representan los datos para una mejor comprensión. Un ejemplo es este gráfico de barras 3D que visualiza un valor objetivo en la dimensión vertical que depende de otros dos valores en las dimensiones horizontales. Puede usar colores para visualizar la tercera dimensión. Otra visualización de datos es la agrupación jerárquica de bordes, que representa las correlaciones y afiliaciones entre entidades. Si es suficiente, un gráfico de barras clásico también puede funcionar. Un gráfico de dispersión 2D con un mapa de calor muestra dos campos de datos dependientes en el eje Y con diferentes intensidades de color. Un mapa de árbol muestra la distribución de los subconjuntos dentro de un conjunto. El famoso gráfico circular hace lo mismo pero de forma no jerárquica. Por último, una nube de palabras muestra los términos significativos de un corpus documental. La construcción de modelos se puede realizar mediante un servicio. Un ejemplo de servicio es Watson Machine Learning. Watson Machine Learning puede entrenar y crear modelos utilizando varias bibliotecas de código abierto. Google tiene un servicio similar en su nube llamado AI Platform Training. Cada proveedor de servicios en la nube tiene una solución para esta tarea. La implementación de modelos en el software comercial suele estar estrechamente integrada en el proceso de creación de modelos. Este es un ejemplo de los servicios de colaboración e implementación de SPSS, que se pueden utilizar para implementar cualquier activo creado por el conjunto de herramientas de software de SPSS. Lo mismo ocurre con otros proveedores. Además, el software comercial puede exportar modelos en un formato abierto. Por ejemplo, SPSS Modeler admite la exportación de modelos en formato Predictive Model Markup Language (PMML), que pueden leer otros paquetes de software comerciales y abiertos. Además, Watson Machine Learning despliega un modelo y lo pone a disposición de los consumidores mediante una interfaz REST. Amazon SageMaker Model Monitor es un ejemplo de una herramienta en la nube para monitorear continuamente los modelos implementados de aprendizaje automático y aprendizaje profundo. Todos los principales proveedores de servicios en la nube tienen herramientas similares. Otra herramienta para la supervisión de modelos es Watson OpenScale. Todo lo que esté marcado en verde se puede hacer con Watson Studio y OpenScale. En este vídeo, ha aprendido: Watson Studio y Watson OpenScale cubren el ciclo de vida completo del desarrollo de todas las tareas de ciencia de datos, aprendizaje automático e inteligencia artificial. En la gestión de datos, con algunas excepciones, existe una versión de software como servicio (SaaS) de las herramientas comerciales y de código abierto existentes. Dos herramientas de integración de datos comerciales muy utilizadas son Informatica Cloud Data Integration y Data Refinery de IBM. Un ejemplo de una herramienta de visualización de datos basada en la nube es Datameer y la suite de inteligencia empresarial Cognos de IBM. La creación de modelos se puede realizar mediante un servicio como Watson Machine Learning. Amazon SageMaker Model Monitor es un ejemplo de una herramienta en la nube para monitorear continuamente los modelos implementados de aprendizaje automático y aprendizaje profundo.**

**Objetivo**

Esta lectura proporciona un resumen de las herramientas clave de código abierto para la ciencia de datos cubiertas en los videos de la Parte 1 y la Parte 2 de este curso.

Se clasifican ampliamente como:

Herramientas de gestión de datos: facilita el almacenamiento, la organización y la recuperación de datos. Incluye Bases de Datos Relacionales, Bases de Datos NoSQL y plataformas Big Data.

Herramientas de transformación e integración de datos: agiliza los canales de datos y automatiza los flujos de trabajo de procesamiento de datos. La tarea de integración y transformación de datos en el mundo clásico del almacenamiento de datos es extraer, transformar y cargar (ETL).

Herramientas de visualización de datos: proporciona representación gráfica de datos y ayuda a comunicar conocimientos.

Herramientas de implementación, monitoreo y evaluación de modelos: admite la creación, implementación, monitoreo y evaluación de datos y modelos de aprendizaje automático.

Herramientas de gestión de activos de datos: organiza y gestiona datos, aplica controles de acceso y garantiza copias de seguridad de activos.

Herramientas de ejecución y desarrollo de código: proporciona entornos para desarrollar, probar e implementar código, ofreciendo recursos computacionales para ejecutarlo.

Herramientas de gestión de activos de código: permite el almacenamiento y la gestión de código, realiza un seguimiento de los cambios y respalda el desarrollo colaborativo.

Herramientas de gestión de datos

mysql

Popular sistema de gestión de bases de datos relacionales de código abierto (RDBMS)

Utiliza lenguaje de consulta estructurado (SQL) para administrar y almacenar datos.

Usos comunes:

aplicaciones web

almacenamiento de datos

Comerc**io electrónico**

**PostgreSQL**

Sistema de gestión de bases de datos relacionales (RDBMS) potente y de código abierto

Enfatiza la extensibilidad y el cumplimiento de SQL.

Ofrece funciones avanzadas como:

Soporte para JSON

Búsqueda de texto completo

Datos espaciales

**Apache CouchDB**

Base de datos NoSQL orientada a documentos

Utiliza JSON para almacenar datos.

Altamente escalable

Tolerante a fallos

Fácil de usar

**MongoDB**

Base de datos NoSQL orientada a documentos

Almacena datos en un JSON flexible

Proporciona:

Escalabilidad

Alta disponibilidad

Distribución de datos

Adecuado para aplicaciones web modernas que manejan grandes volúmenes de datos no estructurados.

**apache casandra**

Base de datos NoSQL orientada a documentos, distribuida y altamente escalable

Puede manejar grandes cantidades de datos estructurados y no estructurados en muchos servidores básicos.

Ofertas:

Alta disponibilidad

Tolerancia a fallos

Niveles de consistencia ajustables

Adecuado para aplicaciones de misión crítica

**Sistema de archivos distribuido Hadoop (HDFS)**

Diseñado para trabajar con grandes conjuntos de datos como Apache Hadoop en un entorno informático distribuido

Procesamiento de datos de alto rendimiento mediante la división de archivos en bloques (predeterminado 128 MB), y estos bloques se distribuyen en múltiples DataNodes

Los datos se replican en diferentes DataNodes, lo que garantiza una alta disponibilidad y tolerancia a fallos.

Escalable y eficiente

**cef**

Plataforma de almacenamiento definida por software gratuita y de código abierto adecuada para entornos de nube híbrida

Diseñado para centros de datos modernos

Proporciona un sistema de almacenamiento unificado altamente escalable que se puede utilizar para almacenamiento de objetos (como AWS S3), almacenamiento en bloques (como discos virtuales para máquinas virtuales) y almacenamiento de archivos (como NFS) en un sistema unificado.

Alto rendimiento, disponibilidad y confiabilidad

**búsqueda elástica**

Principalmente un motor de búsqueda RESTful distribuido y una herramienta de análisis.

Basado en la biblioteca Lucene.

Búsqueda de texto completo, análisis de datos en tiempo real

Altamente escalable

Fácil de usar

Potentes capacidades de consulta

Indexación de datos en tiempo real para una rápida recuperación de documentos.

Herramientas de integración y transformación de datos

**Flujo de aire Apache**

Plataforma de código abierto para crear, programar y monitorear flujos de trabajo mediante programación

Creado originalmente por Airbnb

Permite a los usuarios definir y ejecutar flujos de trabajo complejos

Soporte para:

Dependencias de tareas

Paralelismo

Manejo de errores

**Kubeflow**

Un conjunto de herramientas de aprendizaje automático de código abierto que permite la ejecución de procesos de ciencia de datos sobre Kubernetes.

Proporciona una plataforma para crear, implementar y gestionar flujos de trabajo de aprendizaje automático de un extremo a otro a escala.

Soporte para:

Entrenamiento distribuido

Servicio modelo

Ajuste de hiperparámetros

**Apache Kafka**

Plataforma de transmisión distribuida que permite a las aplicaciones publicar, procesar y suscribirse a secuencias de registros en tiempo real.

Creado originalmente en LinkedIn.

Es escalable, tolerante a fallos y de alto rendimiento.

Adecuado para crear aplicaciones de misión crítica con uso intensivo de datos

**Apache Ni-Fi**

Una plataforma de integración de datos de código abierto que permite a los usuarios automatizar el flujo de datos entre sistemas.

Proporciona una interfaz de usuario basada en web para diseñar y gestionar flujos de datos.

Soporte para:

Enrutamiento de datos

Transformación

Enriquecimiento

Entre otras capacidades

**Apache Spark SQL**

Un módulo en el ecosistema Spark que proporciona una interfaz de programación para trabajar con datos estructurados usando:

SQL

marcos de datos

Conjuntos de datos

Admite una amplia gama de fuentes de datos y proporciona un rendimiento optimizado para tareas complejas de procesamiento de datos.

**Node-RED**

Una herramienta de programación visual de código abierto para conectar dispositivos de hardware, API y servicios en línea.

Permite a los usuarios crear flujos de mensajes basados ​​en eventos.

Bajo consumo de recursos, incluso se ejecuta en dispositivos pequeños como una Raspberry Pi.

Soporte para:

Transformación de datos

Filtración

Agregación

Herramientas de visualización de datos

**PixieDust**

Biblioteca de código abierto para crear visualizaciones de datos exploratorias e interactivas en cuadernos de Python y Jupyter.

Proporciona una variedad de visualizaciones y conectores de datos integrados.

Soporte para personalización y extensibilidad a través de bibliotecas de terceros.

**HUE**

Interfaz web de código abierto para analizar y visualizar grandes conjuntos de datos en Apache Hadoop

Ofrece una experiencia fácil de usar para explorar datos y crear visualizaciones.

No se necesitan conocimientos de programación; Puede crear visualizaciones a partir de consultas SQL.

**kibana**

Herramienta de visualización de datos de código abierto que permite a los usuarios interactuar con sus datos a través de una interfaz basada en web.

Comúnmente utilizado con Elasticsearch para analizar y visualizar grandes conjuntos de datos.

**Apache Superset**

Una aplicación web de inteligencia empresarial moderna y preparada para empresas que facilita la visualización y exploración de grandes conjuntos de datos.

Ofrece un amplio conjunto de opciones de visualización de datos, que incluyen:

Gráficos

Mesas

Mapas

Análisis geoespacial

Procesamiento de datos en tiempo real

Herramientas de implementación de modelos

Predicción ApacheIO (**Apache PredictionIO)**

Servidor de aprendizaje automático de código abierto construido sobre una infraestructura escalable y distribuida

Permite a los desarrolladores crear, evaluar e implementar rápidamente motores predictivos para diversos casos de uso, como:

Recomendación

Clasificación

Agrupación

**Kubernetes**

Plataforma de código abierto para la orquestación de contenedores

Lanza, escala y gestiona automáticamente aplicaciones en contenedores

Ofreciendo características como:

Escalado automático

Autosanación

Equilibrio de carga

Permite la gestión y orquestación de contenedores en numerosos hosts.

**Apache Seldon**

Plataforma de código abierto para implementar y gestionar modelos de aprendizaje automático en Kubernetes

Proporciona una manera de:

Sirve modelos a escala

Automatizar los flujos de trabajo de implementación de modelos

Supervise el rendimiento de los modelos implementados en tiempo real

MLeap

Biblioteca de código abierto para serializar y deserializar modelos de aprendizaje en un archivo multiplataforma

Brinda a los usuarios la capacidad de exportar modelos desde diferentes bibliotecas y marcos de aprendizaje automático, como:

Chispa

Aprendizaje científico

TensorFlow

Los implementa en entornos de producción de alto rendimiento y baja latencia.

TensorFlow Lite

Herramienta de código abierto para ejecutar modelos de aprendizaje automático en dispositivos móviles e integrados

Permite una inferencia efectiva en plataformas móviles e integradas.

Admite una variedad de aceleradores de hardware como:

CPU

GPU

ASIC personalizados

Red Hat OpenShift

Marco de aplicación de contenedor basado en Kubernetes

Con características como automatización, escalabilidad y seguridad.

Ofrece un método para crear, implementar y administrar aplicaciones en contenedores.

Servicio de TensorFlow

Utilidad de código abierto que sirve modelos de aprendizaje automático en entornos del mundo real.

Admite interfaces HTTP y gRPC para ofrecer predicciones

Proporciona alta escalabilidad y baja latencia, implementación y gestión de modelos de TensorFlow.

TensorFlow.js

Biblioteca de código abierto para crear e implementar modelos de aprendizaje automático en JavaScript

Le permite he y ejecutar modelos directamente en el navegador o en Node.js

Admite una amplia gama de arquitecturas de modelos, incluidas redes neuronales, árboles de decisión y

Herramientas de seguimiento y evaluación del modelo

ModeloDB

Plataforma de código abierto para gestionar modelos y experimentos de aprendizaje automático.

Proporciona una manera de rastrear y reproducir experimentos, versionar modelos y colaborar con los miembros del equipo.

Prometeo

Sistema de monitoreo disponible gratuitamente que recopila y almacena métricas en tiempo real de diferentes fuentes.

Le permite visualizar y configurar alertas sobre el estado y el rendimiento de sistemas y aplicaciones.

Admite una variedad de métodos de recopilación de datos, como puntos finales HTTP, exportadores y agentes

Equidad de IA de IBM 360

Conjunto de herramientas de código abierto para detectar y mitigar sesgos en modelos de aprendizaje automático

Proporciona una manera de medir la equidad y el sesgo de los modelos, así como un conjunto de algoritmos para mitigar el sesgo y crear modelos más justos.

Explicabilidad de IBM AI 360

Conjunto de herramientas de código abierto para explicar el comportamiento y las decisiones de los modelos de aprendizaje automático

Proporciona una forma de medir la explicabilidad e interpretabilidad de los modelos, así como un conjunto de algoritmos para generar explicaciones y visualizaciones.

Caja de herramientas IBM Adversarial Robustness 360

Biblioteca gratuita y de código abierto para proteger los modelos de aprendizaje automático de ataques adversarios

Incluye un método para medir la robustez y la vulnerabilidad del modelo.

Incluye un conjunto de algoritmos para mejorar la solidez del modelo y detectar ejemplos contradictorios.

Herramientas de desarrollo y ejecución de código

IDE de Jupyter

Esfuerzo de código abierto

Soporta:

Julia

Pitón

Desarrollo R con Jupyter Notebook

JupyterLab

Jupyter Hub

Cree y comparta documentos que contengan:

código en vivo

Ecuaciones

Visualizaciones

texto narrativo

JupyterLab incluye:

Organización de cuadernos personalizados

JupyterHub extiende todas estas capacidades a la empresa

RStudio

Para desarrolladores

IDE gratuito y de código abierto

Creado para gestionar y ejecutar código R

Funciona en todas las plataformas

Incluye:

Control de versiones

Capacidades de gestión de proyectos

Microsoft Visual Studio

Un IDE que admite una variedad de lenguajes de programación, que incluyen:

do

C++

C++/CLI

Visual Basic.NET

DO#

F#

javascript

Mecanografiado

XML

XSLT

HTML

CSS

Usando complementos, admite:

Pitón

Rubí

Nodo.js

METRO

Otros idiomas

PyCharm

Principalmente un entorno IDE basado en suscripción

Ofrece más de 16 herramientas adicionales para asistencia con codificación, pruebas y desarrollo web.

Apoya el desarrollo científico con integración de IPython y compatibilidad con Matplotlib y NumPy

También ofrece un IDE gratuito de código abierto y basado en la comunidad con capacidades limitadas.

espía

IDE gratuito y de código abierto basado en Python, diseñado por y para científicos, ingenieros y analistas de datos.

Presenta una combinación única de herramientas de desarrollo integrales para:

Edición avanzada

Análisis

Depuración

Perfilado

Capacidades de visualización

Navegador Anaconda

Navegador basado en GUI de código abierto que admite el desarrollo de Python y se integra con:

Eclipse y PyDev

INACTIVO

IntelliJ

Código de Microsoft Visual Studio (código VS)

IDE ninja

PyCharm

Python para código de Visual Studio

Herramientas Python para Visual Studio (PTVS)

espía

Texto sublime

IDE de ala

Herramientas de gestión de activos de código

git

Sistema de control de versiones de código abierto para rastrear cambios en el código y colaboración entre desarrolladores.

Proporciona una forma de gestionar y organizar cambios de código, colaborar en el desarrollo de código y mantener un historial de revisiones de código.

GitLab

Administrador de repositorios Git basado en web

Proporciona una plataforma DevOps completa para:

Gestión de código fuente

Integración e implementación continuas

Escucha

Permite a los equipos colaborar en:

Desarrollo de código

Automatizar los procesos de construcción e implementación

Realice un seguimiento de las métricas y el rendimiento durante todo el ciclo de vida del desarrollo de software.

GitHub

Servicio de alojamiento de repositorios Git basado en web que proporciona una plataforma para que los desarrolladores colaboren en el código y administren proyectos de software.

Permite a los usuarios:

Crear, bifurcar y contribuir a proyectos de código abierto.

Seguimiento de cambios en el código

Gestionar problemas

Solicitudes de extracción

Bitbucket de Atlassian

Servicio de alojamiento de repositorios Git basado en web

Proporciona una plataforma para que los desarrolladores colaboren en el código y administren proyectos de software, con características como:

Solicitudes de extracción

Revisión de código

Permisos de sucursal

**Resumen del módulo 1**

**Estado: Traducido automáticamente del Inglés**

**Traducido automáticamente del Inglés**

¡Enhorabuena! Ha completado este módulo. A estas alturas del curso, ya lo sabe:

* Las categorías de tareas de la ciencia de datos incluyen:
  + Gestión de datos - almacenamiento, gestión y recuperación de datos
  + Integración y transformación de datos - agilizar los conductos de datos y automatizar las tareas de procesamiento de datos
  + Visualización de datos - proporcionan una representación gráfica de los datos y ayudan a comunicar las percepciones
  + Modelización - permiten construir, desplegar, supervisar y evaluar modelos de datos y de aprendizaje automático
* Las tareas de ciencia de datos dan soporte a lo siguiente
  + Gestión de activos de código - almacenar y gestionar el código, realizar un seguimiento de los cambios y permitir el desarrollo colaborativo
  + Gestión de activos de datos - organizar y gestionar los datos, proporcionar control de acceso y realizar copias de seguridad de los activos
  + Entornos de desarrollo - desarrollan, prueban y despliegan el código
  + Entornos de ejecución - proporcionan recursos computacionales y ejecutan el código

El ecosistema de la ciencia de datos consta de muchas opciones comerciales y de código abierto, e incluye tanto aplicaciones tradicionales de escritorio como herramientas basadas en servidores, así como servicios basados en la nube a los que se puede acceder mediante navegadores web e interfaces móviles.

**Herramientas de gestión de datos**: incluyen bases de datos relacionales, bases de datos NoSQL y plataformas de Big Data:

* MySQL y PostgreSQL son ejemplos de sistemas de gestión de bases de datos relacionales (RDBMS) de código abierto, e IBM Db2 y SQL Server son ejemplos de RDBMS comerciales y también están disponibles como servicios en la nube.
* MongoDB y Apache Cassandra son ejemplos de bases de datos NoSQL.
* Apache Hadoop y Apache Spark se utilizan para el análisis de Big Data.

Herramientas de**integración y transformación** de datos: incluyen Apache Airflow y Apache Kafka.

Herramientas de**visualización** de datos: incluyen ofertas comerciales como Cognos Analytics, Tableau y PowerBI y pueden utilizarse para construir cuadros de mando dinámicos e interactivos.

**Herramientas de Flujo:** Git es una herramienta esencial de gestión de activos de código. GitHub es una popular plataforma basada en web para almacenar y gestionar el código fuente. Sus características la convierten en una herramienta ideal para el desarrollo colaborativo de software, incluyendo el control de versiones, el seguimiento de incidencias y la gestión de proyectos.

**Entornos de desarrollo:** Los entornos de desarrollo más populares para la Ciencia de Datos incluyen Jupyter Notebooks y RStudio.

* Jupyter Notebooks proporciona un entorno interactivo para crear y compartir código, texto descriptivo, visualizaciones de datos y otros artefactos computacionales en una interfaz basada en un navegador web.
* RStudio es un entorno de desarrollo integrado (IDE) diseñado específicamente para trabajar con el lenguaje de programación R, que es una herramienta popular para la computación estadística y el análisis de datos.

Lenguaje de la ciencia de datos.

Bienvenido a Languages of Data Science. Después de ver este vídeo, podrá identificar los criterios para determinar el idioma que debe aprender y enumerar las funciones que implica la ciencia de datos. Para los usuarios que recién comienzan su andadura en la ciencia de datos, la gama de opciones técnicas disponibles puede resultar abrumadora. Hay literalmente miles de lenguajes de programación diferentes con sus propias fortalezas y debilidades. Entonces, ¿qué idioma debes aprender primero? La respuesta a esa pregunta depende en gran medida de tus necesidades, de los problemas que intentes resolver y de para quién los resuelvas. En primer lugar, recomendamos considerar los lenguajes Python, R y SQL. Sin embargo, también son populares otros lenguajes como Scala, Java, C++ y Julia con características específicas. JavaScript, PHP, Go, Ruby y Visual Basic también tienen sus propios casos de uso únicos. El idioma que elijas aprender dependerá de lo que necesites lograr y de los problemas que necesites resolver. Los problemas pueden estar relacionados con la empresa para la que trabaja, el puesto que desempeña y la antigüedad de su solicitud actual. Entonces, ¿qué problemas necesitas resolver? Conoceremos mejor las respuestas a esta pregunta a medida que aprendamos más sobre los lenguajes populares en la industria de la ciencia de datos. Hay muchos puestos disponibles para las personas que estén interesadas en dedicarse a la ciencia de datos, entre los que se incluyen: analista de negocios, ingeniero de bases de datos, analista de datos, ingeniero de datos, científico de datos , científico de investigación , ingeniero de software , estadístico, director de productos, director de proyectos, etc. En este vídeo, aprendió que debe seleccionar un idioma para aprender en función de sus necesidades, los problemas que intenta resolver y para quién los está resolviendo. Los lenguajes más populares son Python, R, SQL, Scala, Java, C++ y Julia. JavaScript, PHP, Go, Ruby y Visual Basic también tienen sus propios casos de uso únicos. Los problemas que debe resolver pueden estar relacionados con su empresa, su función y la antigüedad de la aplicación existente.

Introducción a Pyton.

Bienvenido a «Introducción a Python». Después de ver este vídeo, podrá identificar a los usuarios de Python. Enumere los beneficios de usar Python. Describa los esfuerzos de diversidad e inclusión de la comunidad de Python. Python es la fuente inagotable de un lenguaje. Es el lenguaje de programación más utilizado y popular en la industria de la ciencia de datos. Según la encuesta de Kaggle sobre ciencia de datos y aprendizaje automático de 2019, ¾ de los más de 10 000 encuestados de todo el mundo declararon que utilizan Python con regularidad. Glassdoor informó que, en 2019, más del 75% de los puestos de ciencia de datos listados incluían Python en sus descripciones de trabajo. Cuando se les pregunta qué idioma debería aprender primero un aspirante a científico de datos, la mayoría de los científicos de datos responden a Python. Empecemos por las personas que usan Python. Si ya sabe programar, Python es ideal para usted porque utiliza una sintaxis clara y legible. Puedes desarrollar los mismos programas desde otros lenguajes con menos código usando Python. Para los principiantes, Python es un buen lenguaje para empezar debido a la enorme comunidad global y a la gran cantidad de documentación. Varias encuestas diferentes realizadas en 2019 establecieron que más del 80% de los profesionales de datos utilizan Python en todo el mundo. Python es útil en muchas áreas, como la ciencia de datos, la inteligencia artificial y el aprendizaje automático, el desarrollo web y los dispositivos de Internet de las cosas (IoT), como la Raspberry Pi. Entre las grandes organizaciones que utilizan mucho Python se encuentran IBM, Wikipedia, Google y Yahoo! , el CERN, la NASA, Facebook, Amazon, Instagram, Spotify y Reddit. Python cuenta con un amplio apoyo de una comunidad global y está dirigido por la Python Software Foundation. Python es un lenguaje de programación de alto nivel y de uso general que se puede aplicar a muchos tipos diferentes de problemas. Cuenta con una gran biblioteca estándar que proporciona herramientas adecuadas para muchas tareas diferentes, incluidas, entre otras, las bases de datos, la automatización, la extracción de datos web, el procesamiento de textos, el procesamiento de imágenes, el aprendizaje automático y el análisis de datos. Para la ciencia de datos, puede utilizar las bibliotecas de computación científica de Python, como Pandas, NumPy, SciPy y Matplotlib. Para la inteligencia artificial, tiene TensorFlow, PyTorch, Keras y Scikit-learn. Python también se puede usar para el procesamiento del lenguaje natural (NLP) mediante el kit de herramientas de lenguaje natural (NLTK). Otro gran argumento de venta de Python es que la comunidad de Python tiene un historial bien documentado de allanar el camino para los esfuerzos de diversidad e inclusión en la industria tecnológica en su conjunto. El lenguaje Python tiene un código de conducta ejecutado por la Python Software Foundation que busca garantizar la seguridad y la inclusión para todos, tanto en las comunidades Python en línea como en las presenciales. Comunidades como PyLadies buscan crear espacios para las personas interesadas en aprender Python en entornos seguros e inclusivos. PyLadies es un grupo internacional de mentores que se centra en ayudar a más mujeres a convertirse en participantes activas y líderes en la comunidad de código abierto de Python. En este vídeo, aprendiste que Python usa una sintaxis clara y legible. Python tiene una enorme comunidad global y una gran cantidad de documentación. Para la ciencia de datos, puedes usar las bibliotecas de computación científica de Python, como Pandas, NumPy, SciPy y Matplotlib. Python también se puede utilizar para el procesamiento del lenguaje natural (NLP) mediante el kit de herramientas de lenguaje natural (NLTK). La comunidad de Python tiene un historial bien documentado de allanar el camino para los esfuerzos de diversidad e inclusión en la industria tecnológica en su conjunto.

Introducción al lenguaje R

Bienvenido a «Introducción al lenguaje R». Después de ver este vídeo, podrá: comparar el código abierto con el software libre, identificar a los usuarios del lenguaje R, enumerar las ventajas de usar R y enumerar las comunidades globales para conectarse con otros usuarios de R. Según los resultados de la encuesta de 2019 de Kaggle Data Science, en la que participaron más de diez mil personas en todo el mundo, ¡aprender tres idiomas puede suponer una ventaja adicional! Y R ofrece muchas posibilidades. Ahora, Python es de código abierto, mientras que R es software libre. Comparemos el código abierto y el software libre. Las similitudes son que ambos son de uso gratuito. Por lo general, ambos se refieren al mismo conjunto de licencias. Por ejemplo, muchos proyectos de código abierto utilizan la Licencia Pública General (GNU). Ambos apoyan la colaboración. Y en muchos casos, estos términos se pueden usar indistintamente (pero no en todos). Ahora, analicemos las diferencias entre el código abierto y el software libre. La Iniciativa de Código Abierto (OSI) defiende el código abierto, mientras que la Fundación para el Software Libre (FSF) define el software libre. El código abierto se centra más en los negocios, mientras que el software libre se centra más en un conjunto de valores. Entonces, ¿por qué R? Deberías aprender R porque es software libre. Puedes usar el lenguaje de la misma manera que contribuyes al código abierto. Además, permite el uso privado, el uso comercial y la colaboración pública. R es otro idioma respaldado por una amplia comunidad global de personas que desean usar el idioma para resolver grandes problemas. Los estadísticos, los matemáticos y los mineros de datos utilizan R para desarrollar software estadístico, gráficos y análisis de datos. La sintaxis orientada a matrices de R Language facilita la traducción de las matemáticas al código para los estudiantes con una formación mínima o nula en programación. Según la encuesta de ciencia de datos y aprendizaje automático de Kaggle, la mayoría de los programadores aprenden R después de unos años de carrera en ciencia de datos. Y R es muy popular en el mundo académico. Además, entre las empresas que utilizan R se encuentran IBM, Google, Facebook, Microsoft, Bank of America, Ford, TechCrunch, Uber y Trulia. R se ha convertido en el depósito de conocimientos estadísticos más grande del mundo. En 2018, R tenía más de 15 000 paquetes publicados públicamente que permitían realizar complejos análisis exploratorios de datos. R se integra bien con otros lenguajes de programación como C++, Java, C, .Net y Python. Al usar R, las operaciones matemáticas comunes, como la multiplicación de matrices, dan resultados inmediatos. Y R tiene unas prestaciones de programación orientada a objetos más potentes que la mayoría de los lenguajes de computación estadística. Ahora, hay muchas maneras de conectarse con otros usuarios de R en todo el mundo. Para conectarse con otros usuarios, puede utilizar comunidades como uSER, WhyR, SatrDays y R-Ladies. Además, puedes visitar el sitio web del proyecto R para ver las conferencias y eventos de R. En este vídeo, aprendiste que: la Iniciativa de Código Abierto (OSI) defiende el código abierto, mientras que la Fundación para el Software Libre (FSF) define el software libre. Python es de código abierto y R es software libre. La sintaxis orientada a matrices del lenguaje R facilita la traducción de las matemáticas al código para estudiantes sin conocimientos de programación o con pocos conocimientos de programación. Y R se ha convertido en el depósito de conocimientos estadísticos más grande del mundo.

**Introducción a SQL Services**

Bienvenido a «Introducción a SQL». Después de ver este vídeo, podrá: explicar SQL y las bases de datos relacionales, definir los elementos de SQL y enumerar las ventajas del uso de SQL. SQL es un poco diferente a los otros lenguajes. Oficialmente se pronuncia como «ess cue el», aunque algunos lo llaman «secuela». Y aunque el acrónimo significa «lenguaje de consulta estructurado», muchas personas consideran que SQL es diferente de otros lenguajes de desarrollo de software porque no es un lenguaje procedimental. Su ámbito de aplicación se limita a la consulta y la gestión de datos. Si bien no es un lenguaje de «ciencia de datos», ¡ los científicos de datos lo utilizan con regularidad porque es simple y poderoso! Algunos otros datos sobre SQL son que es más antiguo que Python y R en unos 20 años. ¡Apareció por primera vez en 1974 y fue desarrollado en IBM! Este lenguaje es útil para manejar datos estructurados, que son los datos que incorporan relaciones entre entidades y variables. SQL se diseñó para administrar datos en bases de datos relacionales. Aquí puede ver un diagrama que muestra la estructura general de una base de datos relacional. Una base de datos relacional está formada por conjuntos de tablas bidimensionales, por ejemplo, conjuntos de datos y hojas de cálculo de Excel. A continuación, cada una de estas tablas está formada por un número fijo de columnas y cualquier número posible de filas. Sin embargo, aunque SQL se desarrolló originalmente para su uso con bases de datos relacionales, debido a su ubicuidad y facilidad de uso, también se han desarrollado interfaces SQL para muchos repositorios NoSQL y de macrodatos. El lenguaje SQL se subdivide en varios elementos del lenguaje, que incluyen: cláusulas, expresiones, predicados, consultas y declaraciones. Entonces, ¿qué hace que SQL sea excelente? Conocer SQL le ayudará a conseguir muchos trabajos diferentes en el campo de la ciencia de datos, como el de analista empresarial y de datos. Este conocimiento también es imprescindible en la ingeniería de datos. Al realizar operaciones con SQL, se accede a los datos directamente, sin necesidad de copiarlos por separado, lo que puede acelerar considerablemente la ejecución del flujo de trabajo. SQL actúa como un intérprete entre usted y la base de datos. SQL es un estándar del Instituto Nacional de Normalización de los Estados Unidos (o ANSI), lo que significa que si aprende SQL y lo utiliza con una base de datos, podrá aplicar sus conocimientos de SQL a muchas otras bases de datos con facilidad. Ahora hay disponibles muchas bases de datos SQL diferentes, incluidas las siguientes: MySQL, IBM DB2, PostgreSQL, Apache Open Office Base, SQLite, Oracle, MariaDB, Microsoft SQL Server y más. La sintaxis del SQL que escriba puede cambiar en función del sistema de administración de bases de datos relacionales que utilice. Si quiere aprender SQL, debe centrarse en una base de datos relacional específica y, después, conectarse a la comunidad de esa plataforma específica. Además, ¡hay muchos excelentes cursos de introducción a SQL disponibles! En este vídeo, aprendió lo siguiente: SQL se diferencia de otros lenguajes de desarrollo de software porque no es un lenguaje procedimental. El alcance de SQL se limita a la consulta y la administración de datos. SQL se diseñó para administrar datos en bases de datos relacionales. SQL se comporta como un intérprete entre usted y la base de datos. Y si aprende SQL y lo usa con una base de datos, podrá aplicar sus conocimientos de SQL a muchas otras bases de datos fácilmente.

Otros lenguajes para la ciencia de datos

Bienvenido a «Otros lenguajes para la ciencia de datos». Después de ver este vídeo, podrá revisar otros lenguajes, como Java, Scala, C++, JavaScript y Julia, y explorar cómo se usa cada uno de ellos en la ciencia de datos. Anteriormente, analizamos Python, R y SQL. En esta lección, analizaremos otros lenguajes que tienen casos de uso convincentes para la ciencia de datos. Scala, Java, C++ y Julia son probablemente los lenguajes de ciencia de datos más tradicionales. Sin embargo, JavaScript, PHP, Go, Ruby, Visual Basic y muchos otros han encontrado su lugar en la comunidad de la ciencia de datos. Repasemos algunos aspectos destacados de algunos de ellos. Java es un lenguaje de programación orientado a objetos de uso general probado y comprobado. Tiene una gran adopción en el ámbito empresarial y se diseñó para ser rápido y escalable. Las aplicaciones Java se compilan en código de bytes y se ejecutan en la máquina virtual Java o JVM. Algunas herramientas de ciencia de datos destacadas creadas con Java incluyen: Weka para la minería de datos, Java-ML para el aprendizaje automático, Apache MLlib hace que el aprendizaje automático sea escalable y Deeplearning4 para el aprendizaje profundo. Hadoop es otra aplicación de Java que gestiona el procesamiento y el almacenamiento de datos para aplicaciones de macrodatos que se ejecutan en sistemas agrupados. Scala es un lenguaje de programación de uso general que proporciona soporte para la programación funcional y es un sólido sistema de tipo estático. El lenguaje Scala se creó para abordar las deficiencias de Java. También es interoperable con Java, ya que se ejecuta en la JVM. El nombre Scala es una combinación de escalabilidad y lenguaje. Este lenguaje está diseñado para evolucionar con los requisitos de sus usuarios. Para la ciencia de datos, el programa más popular creado con Scala es Apache Spark. Spark es un sistema de computación en clúster rápido y de uso general que proporciona API que facilitan la redacción de trabajos paralelos. Cuenta con un motor optimizado que admite gráficos de cálculo generales. Spark incluye Shark, que es un motor de consultas, MLlib para el aprendizaje automático, GraphX para el procesamiento de gráficos y Spark Streaming. Fue diseñado para ser más rápido que Hadoop. C++ es un lenguaje de programación de uso general. Es una extensión del lenguaje de programación C o «C con clases». C++ mejora la velocidad de procesamiento, permite la programación del sistema y proporciona un control más amplio sobre la aplicación de software. Muchas organizaciones que utilizan Python u otros lenguajes de alto nivel para el análisis de datos y las tareas exploratorias confían en C++ para desarrollar programas que envían datos a los clientes en tiempo real. Para la ciencia de datos, TensorFlow es una popular biblioteca de aprendizaje profundo para flujos de datos que se creó con C++. Aunque C++ es la base de TensorFlow, se ejecuta en una interfaz de Python, por lo que los usuarios no necesitan conocimientos de C++ para ejecutarlo. MongoDB es una base de datos NoSQL para la gestión de macrodatos que se creó con C++. Caffe es un repositorio de algoritmos de aprendizaje profundo creado con C++ con enlaces de Python y Matlab. JavaScript, una tecnología fundamental para la red mundial, es un lenguaje de uso general que se extendió más allá del navegador con la creación de Node.js y otros enfoques del lado del servidor. Javascript NO está relacionado con el lenguaje Java. Para Data Science, sin duda, TensorFlow.js es la implementación más popular. TensorFlow.js hace posible el aprendizaje automático y el aprendizaje profundo en Node.js y en el navegador. TensorFlow.js también fue adoptado por otras bibliotecas de código abierto , como brain.js y machinelearn.js. Otra implementación de JavaScript para la ciencia de datos es R-js. El proyecto R-js ha reescrito las especificaciones de álgebra lineal del lenguaje R para convertirlas en texto mecanografiado. Esto sienta las bases para que los proyectos futuros implementen marcos de base matemática más potentes, como Numpy y SciPy de Python. Typescript es un superconjunto de JavaScript. Por último, Julia fue diseñada en el MIT para el análisis numérico de alto rendimiento y la ciencia computacional. Julia proporciona un desarrollo rápido como Python o R, a la vez que produce programas que se ejecutan tan rápido como los programas en C o Fortran. Está compilado, lo que significa que el código de Julia se ejecuta directamente en el procesador como código ejecutable. Llama a las bibliotecas C, Go, Java, MATLAB, R, Fortran y Python, y tiene un paralelismo refinado. Como lenguaje, Julia solo tiene 8 años, fue escrito en 2012, pero su impacto futuro en la industria de la ciencia de datos es muy prometedor. Una gran aplicación de Julia para la ciencia de datos es JuliaDB, que es un paquete para trabajar con grandes conjuntos de datos persistentes. En este vídeo, descubrirá que las herramientas de ciencia de datos creadas con Java incluyen Weka, Java-ML, Apache MLlib y Deeplearning4. Para la ciencia de datos, un programa popular creado con Scala es Apache Spark, que incluye Shark, MLlib, GraphX y Spark Streaming. Para la ciencia de datos, TensorFlow, MongoDB y Caffe se crearon con C++. Los programas creados para la ciencia de datos con JavaScript incluyen TensorFlow.js y R-js. Una gran aplicación de Julia para la ciencia de datos es JuliaDB.

**Resumen del módulo 2**

**Estado: Traducido automáticamente del Inglés**

**Traducido automáticamente del Inglés**

¡Enhorabuena! Ha completado este módulo. A estas alturas del curso, ya lo sabe:

* Debe seleccionar un lenguaje para aprender en función de sus necesidades, los problemas que intenta resolver y para quién los resuelve.
* Los lenguajes más populares son Python, R, SQL, Scala, Java, C++ y Julia.
* Para la ciencia de datos, puede utilizar las bibliotecas de computación científica de Python como Pandas, NumPy, SciPy y Matplotlib.
* Python también puede utilizarse para el Procesamiento del Lenguaje Natural (PLN) utilizando el Kit de Herramientas de Lenguaje Natural (NLTK).
* Python es de código abierto, y R es software libre.
* La sintaxis orientada a matrices del lenguaje R facilita la traducción de las matemáticas al código a los alumnos sin conocimientos de programación o con conocimientos mínimos.
* SQL es diferente de otros lenguajes de desarrollo de software porque es un lenguaje no procedimental.
* SQL se diseñó para gestionar datos en bases de datos relacionales.
* Si aprende SQL y lo utiliza con una base de datos, podrá aplicar sus conocimientos de SQL con muchas otras bases de datos fácilmente.
* Las herramientas de ciencia de datos construidas con Java incluyen Weka, Java-ML, Apache MLlib y Deeplearning4.
* Para la ciencia de datos, un programa popular construido con Scala es Apache Spark que incluye Shark, MLlib, GraphX, y Spark Streaming.
* Los programas construidos para la Ciencia de Datos con JavaScript incluyen TensorFlow.js y R-js.
* Una gran aplicación de Julia para la Ciencia de Datos es JuliaDB.

**Biblioteca para ciencia de datos**

Las bibliotecas son una colección de funciones y métodos que le permiten realizar una amplia variedad de acciones sin escribir el código usted mismo. Nos centraremos en las bibliotecas de Python: Bibliotecas de Computación Científica en Bibliotecas de Visualización de Python en Python Alto Nivel Machine Learning y Bibliotecas de Deep Learning: «Alto Nivel» simplemente significa que no tiene que preocuparse por los detalles, aunque esto hace que sea difícil estudiar o mejorar Deep Bibliotecas de aprendizaje en Python Libraries usadas en otros idiomas Las bibliotecas generalmente contienen módulos integrados que proporcionan diferentes funcionalidades que puede usar directamente; a veces se llaman «frameworks».

También hay extensas bibliotecas que ofrecen una amplia gama de instalaciones. Pandas ofrece estructuras de datos y herramientas para una limpieza, manipulación y análisis eficaces de datos. Proporciona herramientas para trabajar con diferentes tipos de datos. El instrumento principal de Pandas es una tabla bidimensional que consta de columnas y filas. Esta tabla se denomina «DataFrame» y está diseñada para proporcionar una indexación fácil para que pueda trabajar con sus datos. Las bibliotecas NumPy se basan en matrices, lo que le permite aplicar funciones matemáticas a estas matrices. Pandas se construye en realidad sobre los métodos de visualización de datos NumPy son una gran manera de comunicarse con otros y mostrar los resultados significativos del análisis. Estas bibliotecas le permiten crear gráficos, gráficos y mapas. El paquete Matplotlib es la biblioteca más conocida para la visualización de datos, y es excelente para hacer gráficos y gráficos. Los gráficos también son altamente personalizables. Otra biblioteca de visualización de alto nivel, Seaborn, se basa en matplotlib. Seaborn facilita la generación de parcelas como mapas de calor, series de tiempo y parcelas de violín. Para el aprendizaje automático, la biblioteca Scikit-Learn contiene herramientas para el modelado estadístico, incluyendo regresión, clasificación, clustering y otros. Está construido en NumPy, SciPy y matplotlib, y es relativamente simple comenzar. Para este enfoque de alto nivel, defina el modelo y especifique los tipos de parámetros que desea utilizar. Para el aprendizaje profundo, Keras le permite construir el modelo estándar de aprendizaje profundo. Al igual que Scikit-learn, la interfaz de alto nivel le permite construir modelos de forma rápida y sencilla. Puede funcionar utilizando unidades de procesamiento de gráficos (GPU), pero para muchos casos de aprendizaje profundo se requiere un entorno de nivel inferior

TensorFlow es un marco de bajo nivel utilizado en la producción a gran escala de modelos de aprendizaje profundo. Está diseñado para la producción, pero puede ser difícil de manejar para la experimentación. Pytorch se utiliza para la experimentación, lo que facilita a los investigadores probar sus ideas Apache Spark es un marco de computación de clústeres de propósito general que le permite procesar datos utilizando clústeres informáticos. Esto significa que procesa los datos en paralelo, utilizando varios equipos simultáneamente. La biblioteca Spark tiene una funcionalidad similar a Pandas Numpy Scikit-Learn Apache Spark trabajos de procesamiento de datos pueden usar Python R Scala, o SQL Hay muchas bibliotecas para Scala, que se utiliza predominantemente en ingeniería de datos, pero también a veces se usa en ciencia de datos. Vamos a discutir algunas de las bibliotecas que son complementarias a Spark Vegas es una biblioteca de Scala para visualizaciones de datos estadísticos. Con Vegas, puede trabajar con archivos de datos, así como con Spark DataFrames. Para el aprendizaje profundo, puede usar BigDL. R tiene funcionalidad incorporada para el aprendizaje automático y la visualización de datos, pero también hay varias bibliotecas complementarias: ggplot2 es una biblioteca popular para la visualización de datos en R. También puede usar bibliotecas que le permiten interactuar con Keras y TensorFlow. R ha sido el estándar de facto para la ciencia de datos de código abierto, pero ahora está siendo reemplazado por Python.

**Interfases de programación de aplicaciones (API)**

Bienvenido a las interfaces de programas de aplicaciones o API. Después de ver este video, podrá definir una API, enumerar las bibliotecas de API y definir la API REST en relación con la solicitud y la respuesta. **Una interfaz de programación de aplicaciones, o API**, **permite la comunicación entre dos piezas de software.** Por ejemplo, en un programa tienes algunos datos y otros componentes de software. Utiliza la API para comunicarse mediante entradas y salidas sin saber qué sucede en el back-end. **La API solo se refiere a la interfaz. Es la parte de la biblioteca que ve mientras contiene todos los componentes del programa**. Ahora, para comprender mejor cómo funciona una API en una biblioteca, considere un ejemplo de la biblioteca Pandas. Pandas es un conjunto de componentes de software donde no todos los componentes están escritos en Python. En su programa, hay algunos datos y un conjunto de componentes de software. Puede utilizar la API de Pandas para procesar los datos comunicándose con los otros componentes del software. El componente de software en el back-end puede ser el mismo, pero puede haber una API para diferentes idiomas. Considere TensorFlow en el back-end, escrito en C++, que puede usar API para otros lenguajes, como Python, JavaScript, C++, Java y Go y, por lo tanto, **la API es solo la interfaz.** Otras API desarrolladas por voluntarios para TensorFlow son **Julia, MATLAB, R, Scala y muchas más.** Entonces, las API REST son otro tipo popular de API. La R-E significa Representacional, la S significa Estado y la T significa Transferencia. Le permiten comunicarse a través de Internet y aprovechar recursos, como almacenamiento, datos, algoritmos de inteligencia artificial y mucho más.

En REST API, su programa es el cliente. La API se comunica con un servicio web al que puede llamar a través de Internet, aunque existen reglas con respecto a la comunicación, entrada o solicitud y salida o respuesta. Entonces, veamos algunos términos comunes utilizados con respecto a API. Usted o su código son el cliente. El servicio web es el recurso. El cliente encuentra el servicio a través de un punto final. Y el cliente envía solicitudes al recurso y recibe una respuesta del recurso. Los datos se transmiten a través de Internet mediante métodos HTTP. Las API REST obtienen toda la información de la solicitud enviada por el cliente. La solicitud se envía mediante un mensaje HTTP que contiene un archivo JSON. El archivo contiene instrucciones sobre qué operación debe realizar el servicio web. Esta operación se transmite al servicio web a través de Internet y el servicio realiza la operación. De manera similar, el servicio web devuelve una respuesta a través de un mensaje HTTP donde la información se devuelve mediante un archivo JSON. Y esta información se transmite de vuelta al cliente. Ahora, otro ejemplo de API REST es la API Watson Speech-to-Text. Esta API convierte voz en texto. En la llamada API, enviará una copia del archivo de audio a la API. Esto se llama solicitud de publicación.

En la API, enviaremos la transcripción de texto de lo que dice el individuo. En el back-end, la API realiza una solicitud GET. Y finalmente, veamos nuestro ejemplo final, la API de Watson Language Translator. Envía el texto que desea traducir a la API de Watson Language Translator. La API traducirá el texto y le enviará la traducción. Y en este caso, la API traduce del inglés al español. En este video, aprendió que una interfaz de programación de aplicaciones, o API, permite la comunicación entre dos piezas de software. Una API es la parte de la biblioteca que ve mientras la biblioteca contiene todos los componentes del programa. Y las API REST le permiten comunicarse a través de Internet y aprovechar recursos como almacenamiento, datos, algoritmos de inteligencia artificial y mucho más.

**Conjuntos de datos- impulsar ciencia de datos.**

Bienvenido a «Conjuntos de datos: potenciando la ciencia de datos». Tras ver este vídeo, podrá definir un conjunto de datos, describir los tipos de propiedad de los datos, enumerar las fuentes de datos y describir el acuerdo de licencia de datos comunitarios. Primero definamos qué es un conjunto de datos. Un conjunto de datos es una colección estructurada de datos. Los datos incluyen información representada como texto, números o medios, como archivos de imágenes, audio o vídeo. Un conjunto de datos tabulares comprende una colección de filas que contienen columnas que almacenan la información. Un formato de datos tabulares popular es el de «valores separados por comas» o CSV. Un archivo CSV es un archivo de texto delimitado en el que cada línea representa una fila y una coma separa los valores de los datos. Por ejemplo, imagine un conjunto de datos de observaciones de una estación meteorológica. Cada fila representa una observación en un momento dado, mientras que cada columna contiene información sobre esa observación, como la temperatura, la humedad y otras condiciones meteorológicas. Las estructuras de datos jerárquicas o de red se utilizan normalmente para representar las relaciones entre los datos. Los datos jerárquicos se organizan en un formato similar a un árbol, mientras que los datos de red se almacenan como un gráfico. Por ejemplo, las conexiones entre las personas en un sitio web de redes sociales suelen representarse como un gráfico. Un conjunto de datos también puede incluir archivos de datos sin procesar, como imágenes o audio. El conjunto de datos modificado del Instituto Nacional de Estándares y Tecnología (MNIST) es popular para la ciencia de datos. Contiene imágenes de dígitos manuscritos y se suele utilizar para entrenar sistemas de procesamiento de imágenes. Tradicionalmente, la mayoría de los conjuntos de datos eran privados porque contenían información privada o confidencial, como datos de clientes, datos de precios u otra información confidencial desde el punto de vista comercial. Por lo general, estos conjuntos de datos no se comparten públicamente. Con el tiempo, muchas entidades públicas y privadas, como instituciones científicas, gobiernos, organizaciones e incluso empresas, han empezado a poner los conjuntos de datos a disposición del público como «datos abiertos», lo que proporciona información gratuita. Por ejemplo, las Naciones Unidas y los gobiernos federales y municipales de todo el mundo han publicado muchos conjuntos de datos en sus sitios web, que abarcan la economía, la sociedad, la sanidad, el transporte, el medio ambiente y mucho más. El acceso a estos y otros conjuntos de datos abiertos permite a los científicos de datos, investigadores, analistas y otras personas descubrir información previamente desconocida y potencialmente valiosa. Se utilizan para crear nuevas aplicaciones con fines comerciales y de interés público. También se utilizan para llevar a cabo más investigaciones. Los datos abiertos han desempeñado un papel importante en el crecimiento de la ciencia de datos, el aprendizaje automático y la inteligencia artificial. Ha permitido a los profesionales perfeccionar sus habilidades en varios conjuntos de datos. Hay muchas fuentes de datos abiertas en Internet. Puede encontrar una lista completa de los portales de datos disponibles en todo el mundo en el sitio web datacatalogs.org de la Open Knowledge Foundation. Las Naciones Unidas, la Unión Europea y muchas otras organizaciones gubernamentales e intergubernamentales mantienen repositorios de datos que proporcionan acceso a una amplia gama de información. En Kaggle, una popular comunidad en línea sobre ciencia de datos, puedes encontrar (y aportar) conjuntos de datos que podrían ser de interés general. Google ofrece un motor de búsqueda que puede ayudarte a encontrar conjuntos de datos que podrían serte útiles. La distribución y el uso de los datos abiertos pueden estar restringidos, según lo definen algunos términos de licencia. Sin una licencia para la distribución de datos abiertos, muchos conjuntos de datos se compartían en el pasado bajo licencias de software de código abierto. Estas licencias no se diseñaron para cubrir consideraciones específicas relacionadas con la distribución y el uso de conjuntos de datos. Para solucionar este problema, la Fundación Linux creó el Acuerdo de licencia de datos comunitarios ( CDLA). Inicialmente, se crearon dos licencias para compartir datos: CDLA-Sharing y CDLA-Permissive. La licencia de uso compartido de CDLA le otorga permiso para usar y modificar los datos. La licencia estipula que si publica su versión modificada de los datos, debe hacerlo bajo los mismos términos de licencia que los datos originales. La licencia permisiva de la CDLA también le otorga permiso para usar y modificar los datos. Sin embargo, no está obligado a compartir los cambios en los datos. Tenga en cuenta que ninguna de las dos licencias impone ninguna restricción a los resultados que pueda obtener al utilizar los datos, lo cual es importante en la ciencia de datos. Supongamos, por ejemplo, que está creando un modelo que realiza una predicción. Si está entrenando el modelo con conjuntos de datos con licencia CDLA, no tiene la obligación de compartir el modelo ni de compartirlo bajo una licencia específica si decide compartirlo. En este vídeo, descubrirá que los datos abiertos son fundamentales para la ciencia de datos. El acuerdo de licencia de datos comunitarios facilita el intercambio de datos abiertos, y es posible que los conjuntos de datos abiertos no cumplan con los requisitos empresariales debido al impacto que podrían tener en la empresa.

**Compartir datos empresariales, intercambio de activos de datos**

Bienvenido a compartir datos y datos empresariales. Tras ver este vídeo, podrá navegar por el repositorio de datos abiertos de IBM, el Data Asset Exchange, explorar los conjuntos de datos abiertos en el Data Asset Exchange e identificar el cuaderno asociado a un conjunto de datos en Watson Studio. Ahora hay muchos conjuntos de datos abiertos disponibles para el público, pero puede resultar difícil encontrar conjuntos de datos que sean de alta calidad y que tengan términos de licencia y uso claramente definidos. Para ayudar a resolver este desafío, IBM creó el intercambio de activos de datos o DAX. DAX proporciona una colección seleccionada de conjuntos de datos abiertos tanto de fuentes confiables de terceros como de investigaciones de IBM. Estos conjuntos de datos están listos para usarse en aplicaciones empresariales con una amplia variedad de tipos de aplicaciones que incluyen imágenes, vídeo, texto y audio. El DAX tiene como objetivo fomentar el intercambio de datos y la colaboración al mantener los conjuntos de datos disponibles en virtud de un acuerdo de licencia de datos comunitarios o CDLA. DAX facilita a los desarrolladores la tarea de comenzar con los conjuntos de datos , ya que proporciona un único lugar para acceder a conjuntos de datos únicos y de alta calidad de fuentes confiables como IBM Research. También proporciona cuadernos tutoriales que explican los conceptos básicos de la limpieza de datos, el preprocesamiento y el análisis exploratorio. Algunos conjuntos de datos incluyen cuadernos avanzados que explican cómo realizar tareas más complejas, como la creación de gráficos, el entrenamiento de modelos de aprendizaje automático, la integración del aprendizaje profundo mediante el intercambio de activos de modelos y la ejecución de análisis estadísticos y de series temporales. El Data Asset Exchange y el Model Asset Exchange están disponibles en el sitio web para desarrolladores de IBM. Con estos recursos, los desarrolladores pueden crear flujos de trabajo analíticos y de aprendizaje automático integrales y consumir datos y modelos abiertos con confianza en términos de licencia claramente definidos. En el intercambio de activos de datos, hay varios conjuntos de datos abiertos disponibles para que los explore. Supongamos que ha encontrado un conjunto de datos que puede resultarle muy interesante, los datos meteorológicos de la NOAA, la base de datos del aeropuerto JFK, que contiene datos de una estación meteorológica del aeropuerto John F. Kennedy de Nueva York. En esta página de conjunto de datos, puede hacer clic en obtener este conjunto de datos para descargar el conjunto de datos de la NOAA del almacenamiento en la nube, ejecutar cuadernos de conjuntos de datos para acceder a los cuadernos asociados al conjunto de datos en Watson y obtener una vista previa de los datos y los cuadernos para explorar los metadatos, el glosario y el cuaderno del DAX. Ahora, la mayoría de los conjuntos de datos del DAX se complementan con uno o más cuadernos. Haga clic en los activos para ver todos los cuadernos y datos de Jupyter disponibles. Puedes hacer clic en el código fuente para ver todos los cuadernos asociados a tu proyecto de la NOAA. Puede ejecutar todos los cuadernos de Watson Studio para realizar la limpieza de datos, el preprocesamiento y el análisis exploratorio. Si ya está familiarizado con la apertura de las libretas en Watson Studio, puede iniciar sesión en su cuenta de IBM cloud, crear un proyecto y cargar todas las libretas en el proyecto. Los conjuntos de datos del DAX también constan de uno o más archivos de datos. Haga clic en la opción de datos para ver los archivos de datos disponibles en su proyecto. En este vídeo, descubrió que el sitio de IBM Data Asset Exchange o DAX contiene conjuntos de datos abiertos de alta calidad. Los conjuntos de datos abiertos de DAX incluyen cuadernos tutoriales que proporcionan tutoriales básicos y avanzados para los desarrolladores. DAX y MAX están disponibles en el sitio web para desarrolladores de IBM. Puede ejecutar y previsualizar conjuntos de datos y libretas en DAX y las libretas DAX se abren en Watson Studio.

**Modelos de Aprendizaje automático, Aprender de los modelos para hacer predicciones**

Bienvenido a «Modelos de aprendizaje automático: aprender de los modelos para hacer predicciones». Después de ver este vídeo, podrá definir un modelo de aprendizaje automático, describir los diferentes tipos de modelos de aprendizaje y describir cómo utilizar un modelo de aprendizaje para resolver un problema. Ahora los datos contienen una gran cantidad de información que se puede utilizar para resolver ciertos tipos de problemas. Los enfoques tradicionales de análisis de datos pueden consistir en que una persona inspeccione manualmente los datos o en un programa informático especializado que automatice el análisis humano. Estos enfoques llegan a sus límites debido a la cantidad de datos que se van a analizar o a la complejidad del problema. El aprendizaje automático (ML) utiliza algoritmos, también conocidos como «modelos», para identificar patrones en los datos. El proceso mediante el cual el modelo aprende estos patGITrones a partir de los datos se denomina «entrenamiento con modelos». Una vez que se entrena un modelo, se puede utilizar para hacer predicciones. Cuando se presentan nuevos datos al modelo, intenta hacer predicciones o tomar decisiones en función de los patrones que ha aprendido de los datos anteriores. Los modelos de aprendizaje automático se pueden dividir en tres clases básicas: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo. El tipo de aprendizaje automático más utilizado es el aprendizaje supervisado. En el aprendizaje supervisado, un humano proporciona datos de entrada y resultados correctos. El modelo intenta identificar las relaciones y dependencias entre los datos de entrada y la salida correcta. Este tipo de aprendizaje comprende dos tipos de modelos: regresión y clasificación. Los modelos de regresión se utilizan para predecir un valor numérico (o «real»). Por ejemplo, si se proporciona información sobre las ventas de viviendas anteriores, como la ubicación geográfica, el tamaño, el número de dormitorios y el precio de venta, puede entrenar un modelo para que prediga el precio de venta estimado de otras viviendas con características similares. Los modelos de clasificación se utilizan para predecir si alguna información o dato pertenece a una categoría (o «clase»). Por ejemplo, en el caso de un conjunto de correos electrónicos junto con una designación , puede clasificar si se consideran spam o no. De este modo, puede entrenar un algoritmo para identificar los correos electrónicos no solicitados. En el aprendizaje no supervisado, un humano no etiqueta los datos. Los modelos deben analizar los datos e intentar identificar los patrones y la estructura de los datos en función de sus características. La agrupación es un ejemplo de este estilo de aprendizaje. Los modelos de agrupamiento se utilizan para dividir cada registro de un conjunto de datos en uno de un grupo similar. Un ejemplo de un modelo de agrupamiento podría ser ofrecer recomendaciones de compra para una tienda de comercio electrónico, basadas en el comportamiento de compra anterior y en el contenido de una cesta de la compra. Otro ejemplo es la detección de anomalías, que identifica valores atípicos en un conjunto de datos, como transacciones fraudulentas con tarjetas de crédito o intentos sospechosos de inicio de sesión en línea. Y el tercer tipo de aprendizaje, el aprendizaje por refuerzo, se basa vagamente en la forma en que aprenden los seres humanos y otros organismos. Por lo tanto, piense en un ratón en un laberinto. Si el ratón llega al final del laberinto, obtiene un trozo de queso. Esta es la «recompensa» por completar una tarea. El ratón aprende mediante ensayo y error cómo atravesar el laberinto para conseguir la mayor cantidad de queso posible. De manera similar, un modelo de aprendizaje por refuerzo aprende cuál es el mejor conjunto de acciones a realizar, dado su entorno actual, para obtener la mayor cantidad de recompensas a lo largo del tiempo. Recientemente, este tipo de aprendizaje ha tenido mucho éxito al superar a los mejores jugadores humanos en juegos como el Go, el ajedrez y los populares videojuegos de estrategia. El aprendizaje profundo es un tipo especializado de aprendizaje automático. Se refiere a un conjunto general de modelos y técnicas que emulan vagamente la forma en que el cerebro humano resuelve una amplia gama de problemas. Se usa comúnmente para analizar el lenguaje natural (tanto hablado como textual), imágenes, audio, vídeo, para pronosticar datos de series temporales y mucho más. El aprendizaje profundo ha tenido mucho éxito recientemente en estas y otras áreas y, por lo tanto, se está convirtiendo en una herramienta cada vez más popular e importante para la ciencia de datos. Requiere grandes conjuntos de datos etiquetados para entrenar un modelo, requiere un uso intensivo de cómputos y, por lo general, requiere un hardware específico para lograr tiempos de entrenamiento aceptables. Ahora puede crear un modelo de aprendizaje profundo personalizado desde cero o utilizar modelos previamente entrenados de repositorios de modelos públicos. Los modelos de aprendizaje profundo se implementan mediante marcos populares como TensorFlow, PyTorch y Keras. Los marcos de aprendizaje proporcionan una API de Python y muchos admiten otros lenguajes de programación, como C++ y JavaScript. Puede descargar modelos de última generación previamente entrenados de los repositorios que comúnmente se denominan zoológicos modelo. Entre los zoológicos modelo más populares se incluyen los proporcionados por TensorFlow, PyTorch, Keras y ONNX. Los modelos también son publicados por grupos de investigación académicos y comerciales. Describamos brevemente las tareas de alto nivel que implica la construcción de un modelo con un ejemplo. Suponga que quiere habilitar una aplicación para identificar objetos en imágenes mediante el entrenamiento de un modelo de aprendizaje profundo. En primer lugar, debe recopilar y preparar los datos que se utilizarán para entrenar un modelo. La preparación de los datos puede ser un proceso lento y laborioso. Para entrenar un modelo para que detecte objetos en las imágenes, es necesario etiquetar los datos de entrenamiento sin procesar. Por ejemplo, puede dibujar cuadros delimitadores alrededor de los objetos y etiquetarlos. A continuación, cree un modelo desde cero o seleccione un modelo existente que pueda ser adecuado para la tarea a partir de un recurso público o privado. A continuación, puede entrenar el modelo con los datos que haya preparado. Durante el entrenamiento, el modelo aprende a identificar los objetos representados en una imagen a partir de los datos etiquetados. Una vez que el entrenamiento ha comenzado, usted analiza los resultados del entrenamiento y repite el proceso hasta que el rendimiento del modelo entrenado cumpla con sus requisitos. Cuando el modelo entrenado funciona como se desea, lo despliega para que esté disponible para sus aplicaciones. En este vídeo, aprendió lo siguiente: el aprendizaje automático (ML) utiliza algoritmos, también conocidos como «modelos», para identificar patrones en los datos. El proceso mediante el cual el modelo aprende los patrones de datos se denomina «entrenamiento con modelos». Los tipos de aprendizaje automático son supervisado, no supervisado y reforzado. El aprendizaje supervisado comprende dos tipos de modelos: regresión y clasificación. Y el aprendizaje profundo se refiere a un conjunto general de modelos y técnicas que emulan vagamente la forma en que el cerebro humano resuelve una amplia gama de problemas.

**El intercambio de activos Modelo**

Bienvenido al Model Asset eXchange. Tras ver este vídeo, podrá navegar por el Model Asset eXchange de IBM y explicar cómo el servicio de modelos de aprendizaje profundo detecta las imágenes. El Model Asset eXchange o MAX de la plataforma IBM Developer es un recurso gratuito de código abierto para modelos de aprendizaje profundo. Las tareas necesarias para entrenar un modelo desde cero requieren una gran cantidad de datos, mano de obra, tiempo y recursos, por lo que el tiempo de obtención de valor puede ser bastante largo. Para reducir el tiempo de creación de valor, considere la posibilidad de aprovechar los modelos previamente entrenados para ciertos tipos de problemas. Estos modelos previamente entrenados pueden estar listos para usarse de inmediato o pueden tardar menos en entrenarse. Los modelos se crean mediante la ejecución de datos a través de un modelo utilizando recursos informáticos y experiencia en el dominio. Una vez completados los pasos de investigación, evaluación, prueba, capacitación y validación, tendrá un modelo validado. Model Asset eXchange es un repositorio gratuito de código abierto para microservicios de aprendizaje profundo personalizables y listos para usar. Estos microservicios están configurados para usar modelos de aprendizaje profundo previamente entrenados o personalizados para resolver problemas empresariales comunes. Estos modelos se han probado en su totalidad y se pueden implementar rápidamente en entornos locales y en la nube. Todos los modelos de MAX están disponibles bajo licencias permisivas de código abierto, lo que facilita su uso para fines personales y comerciales, lo que reduce el riesgo de responsabilidades legales. En MAX, puede encontrar modelos para una variedad de dominios, incluida la detección de objetos, la clasificación de imágenes, audio, vídeo y texto, el reconocimiento de entidades con nombre, la traducción de imágenes a texto, la detección de poses humanas y más. Veamos los componentes de un microservicio típico que sirve modelos. Cada microservicio incluye un modelo de aprendizaje profundo previamente entrenado, un código que preprocesa la entrada antes de que el modelo la analice, un código que procesa posteriormente la salida del modelo y una API pública estandarizada que pone la funcionalidad de los servicios a disposición de las aplicaciones. Los microservicios que sirven modelos se crean mediante la ejecución de entradas a través de un modelo validado y, a continuación, la aplicación de la salida a una API REST. Una vez finalizados los pasos de implementación, empaquetado, documentación y prueba, dispondrá de un microservicio que sirve de modelos y que se puede enviar a una máquina local o a una nube pública o híbrida privada. Los microservicios que sirven modelos MAX se crean y distribuyen como imágenes Docker de código abierto. Docker es una plataforma de contenedores que facilita la creación e implementación de aplicaciones. La fuente de imágenes de Docker se publica en GitHub y se puede descargar y personalizar para su uso en entornos personales y comerciales. Utilice el sistema de código abierto de Kubernetes para automatizar la implementación, el escalado y la administración de estas imágenes de Docker. Red Hat OpenShift es una popular plataforma Kubernetes de nivel empresarial. Está disponible en IBM Cloud, Google Cloud Platform, Amazon Web Services y Microsoft Azure. En este vídeo, descubrió que Model Asset eXchange es un repositorio gratuito de código abierto para microservicios de aprendizaje profundo personalizables y listos para usar. Para reducir el tiempo de creación de valor, considere la posibilidad de aprovechar los modelos previamente entrenados para ciertos tipos de problemas. Los microservicios que sirven modelos MAX se crean y distribuyen en GitHub como imágenes Docker de código abierto. Red Hat OpenShift es una plataforma de Kubernetes que se utiliza para automatizar la implementación, el escalado y la administración de microservicios.

Enhorabuena Ha completado este módulo. A estas alturas del curso, ya sabes que:

* Python ofrece un variado ecosistema de librerías para la Ciencia de datos, que abarca la computación científica (Pandas, NumPy), la visualización (Matplotlib, Seaborn) y el Aprendizaje automático de alto nivel (Scikit-learn). Estas bibliotecas ofrecen herramientas para la manipulación de datos, operaciones matemáticas y desarrollo simplificado de modelos de aprendizaje automático.
* Las interfaces de programación de aplicaciones (API) facilitan la comunicación entre componentes de software. Las API REST, en concreto, facilitan la comunicación por Internet y el acceso a recursos como el almacenamiento. Los términos clave de las API son cliente (usuario o código que accede a ella), recurso (servicio o datos) y punto final (URL de la API).
* Los modelos de aprendizaje automático analizan los datos e identifican patrones para hacer predicciones y automatizar tareas complejas: los tres tipos fundamentales de aprendizaje automático son el aprendizaje supervisado, el no supervisado y el de refuerzo. El Aprendizaje supervisado incluye modelos de regresión y clasificación para el modelado predictivo y el reconocimiento de patrones. El Aprendizaje profundo, un subconjunto avanzado del aprendizaje automático, imita el procesamiento cerebral y permite resolver problemas complejos en diversos ámbitos.
* El Acuerdo de Licencia de Datos Comunitarios (CDLA) facilita la puesta en común de datos abiertos proporcionando condiciones de licencia claras para su distribución y uso, y el sitio IBM Data Asset eXchange (DAX) contiene conjuntos de datos abiertos de alta calidad.
* Model Asset eXchange (MAX) proporciona una gran cantidad de modelos de Aprendizaje profundo preentrenados, lo que permite a los desarrolladores desplegar fácilmente soluciones para diversos retos empresariales.

**Introducción al cuaderno de Jupiter**

Bienvenido a la «Introducción a los cuadernos de Jupyter». Tras ver este vídeo, podrá definir un cuaderno de Jupyter, explicar cómo utilizar JupyterLab y describir cómo utilizar los cuadernos de JupyterLab. Los Jupyter Notebooks se originaron como «iPython» y se desarrollaron originalmente para la programación en Python. Más tarde, cuando comenzó a admitir idiomas adicionales, pasó a llamarse Jupyter, que son las siglas de Julia, Python y R. Sin embargo, ahora es compatible con muchos otros lenguajes. A Jupyter Notebook es una aplicación basada en un navegador que permite crear y compartir documentos que contienen código, ecuaciones, visualizaciones, enlaces de texto narrativo y mucho más. Es como el cuaderno de laboratorio de un científico, en el que un científico registra todos los pasos para realizar sus experimentos y los resultados que puede reproducir. Del mismo modo, un cuaderno de Jupyter permite a un científico de datos registrar sus experimentos con datos y los resultados que otros pueden reutilizar. Ahora, un archivo de Jupyter Notebook le permite combinar texto descriptivo , bloques de código y código de salida en un solo archivo. Al ejecutar el código, se genera el resultado, que incluye gráficos y tablas, dentro del archivo del bloc de notas. Y, a continuación, puede exportar el bloc de notas a un formato de archivo PDF o HTML que luego podrá compartir con cualquier persona. A continuación, aprendamos sobre Jupyter Lab. Jupyter Lab es una aplicación basada en un navegador que le permite acceder a varios archivos de Jupyter Notebook, otros códigos y archivos de datos. Además, amplía las funcionalidades de Jupyter Notebooks al permitirle trabajar con varios cuadernos, editores de texto, terminales y componentes personalizados de forma flexible, integrada y ampliable. Es compatible con varios formatos de archivo, como CSV, JSON, PDF, Vega, etc. Y también es de código abierto. Los Jupyter Notebooks se pueden utilizar con servicios basados en la nube, como IBM y Google Colab. No requieren ninguna instalación en su máquina local. Le dan acceso al entorno de Jupyter Notebook y le permiten importar y exportar cuadernos utilizando el formato de archivo estándar de IPython Notebook. Además, estos servicios son compatibles con el lenguaje Python y otros lenguajes. Los Jupyter Notebooks se pueden instalar a través de la línea de comandos mediante la función pip install. También se puede descargar localmente en su portátil a través de la plataforma Anaconda desde Anaconda punto com. Anaconda es una de las distribuciones más populares que incluye Jupyter y Jupyterlab. Por lo tanto, para este curso, tienes acceso a una versión alojada de JupyterLab en Skills Network Labs, por lo que no necesitas ninguna instalación en tu propio dispositivo para completar los laboratorios prácticos. Como se muestra aquí, verás una pantalla que abrirá el Jupyter Lab en el entorno virtual. Simplemente haga clic en la pestaña Abrir herramienta. En este vídeo, descubrirá que los cuadernos de Jupyter se utilizan en la ciencia de datos para registrar experimentos y proyectos. Jupyter Lab es compatible con muchos archivos y lenguajes de ciencia de datos. Y hay diferentes maneras de instalar y usar Jupyter Notebooks.

**Primeros pasos con Jupiter**

Bienvenido a «Primeros pasos con Jupyter». Tras ver este vídeo, podrá: Describir cómo ejecutar, insertar y eliminar una celda de un bloc de notas. Trabaja con varios blocs de notas. Presente el cuaderno y cierre la sesión del cuaderno. En la sesión de laboratorio de este módulo, puede lanzar un cuaderno utilizando el entorno virtual de Skills Network. Tras seleccionar la casilla de verificación, haga clic en la pestaña Abrir herramienta y el entorno abrirá el laboratorio de Jupyter. Aquí puede ver el cuaderno abierto. Al abrir la libreta, puede cambiarle el nombre. Haga clic en Archivo. A continuación, haga clic en Cambiar nombre del bloc de notas para asignar el nombre deseado. Y ahora puede empezar a trabajar en su nueva libreta. En el nuevo cuaderno, imprima «hola mundo». Luego haga clic en el botón Ejecutar para mostrar que el entorno está dando el resultado correcto. En la barra de menú principal de la parte superior, haz clic en Ejecutar. En el menú desplegable, haga clic en Ejecutar celdas seleccionadas para ejecutar las celdas resaltadas actualmente. Como alternativa, puede usar un atajo y presionar Shift + Enter. Si tiene varias celdas de código, haga clic en Ejecutar todas las celdas para ejecutar el código en todas las celdas. Puede añadir código insertando una celda nueva. Para añadir una celda nueva, haga clic en el símbolo más de la barra de herramientas. Además, puede eliminar una celda. Resalte la celda y, en la barra de menú principal, haga clic en Editar y, a continuación, en Eliminar celdas. Como alternativa, puede utilizar un método abreviado pulsando D dos veces en la celda resaltada. Además, puede mover las celdas hacia arriba o hacia abajo según sea necesario. Entonces, ahora has aprendido a trabajar con un solo cuaderno. A continuación, aprendamos a trabajar con varios cuadernos. Haga clic en el botón con el signo más de la barra de herramientas y seleccione el archivo que desee abrir. Se abrirá otro bloc de notas. También puede hacer clic en Archivo en la barra de menús y luego en Abrir un nuevo lanzador o Abrir un bloc de notas nuevo. Y cuando abras el nuevo archivo, podrás moverlo de un lugar a otro. Por ejemplo, como se muestra, puede colocar las libretas una al lado de la otra. En una libreta, puede asignar la variable uno al número 1 y la variable dos al número 2 y, a continuación, imprimir el resultado de sumar los números uno y dos. Como científico de datos, es importante comunicar los resultados. Jupyter permite presentar los resultados directamente desde los cuadernos. Puede crear un Markdown para añadir títulos y descripciones de texto que faciliten el flujo de la presentación. Para añadir Markdown, haz clic en Código y selecciona Markdown. Puede crear gráficos de líneas y convertir cada celda y salida en una diapositiva o diapositiva secundaria en forma de presentación. La funcionalidad de diapositivas de Jupyter te permite entregar código, visualización, texto y resultados del código ejecutado como parte de un proyecto. Ahora, cuando haya terminado de trabajar con su cuaderno o cuadernos, podrá cerrarlos. Al apagar los cuadernos, se libera su memoria. Haz clic en el icono de parada de la barra lateral, es el segundo icono de la parte superior. Puede finalizar todas las sesiones a la vez o cerrarlas individualmente. Y después de cerrar la sesión del bloc de notas, verá «sin núcleo» en la parte superior derecha. Esto confirma que ya no está activo. Ahora puede cerrar las pestañas. En este vídeo, aprendió a: Ejecutar, eliminar e insertar una celda de código. Ejecute varios blocs de notas al mismo tiempo. Presente un cuaderno utilizando una combinación de Markdown y celdas de código. Y cierre las sesiones de su cuaderno una vez que haya completado su trabajo.

**Laboratorio práctico: Introducción a los cuadernos Jupyter**

**Estado: Traducido automáticamente del Inglés**

**Traducido automáticamente del Inglés**

Información:

Este elemento incluye contenido que aún no se tradujo a tu idioma preferido.

Después de completar este laboratorio, usted será capaz de:

* Obtener una visión general de las celdas de Código y Markdown
* Ejecutar una celda de código existente
* Insertar y eliminar una celda de código
* Escribir comentarios en Python
* Crear y utilizar celdas Markdown

Skills Network Labs (SN Labs) es un entorno de laboratorio virtual utilizado en este curso. Al hacer clic en "Abrir herramienta", de acuerdo con la política de privacidad de IBM Skills Network, su nombre de usuario y su dirección de correo electrónico se transmitirán a Skills Network Labs y sólo se utilizarán para comunicarle información importante para mejorar su experiencia de aprendizaje.

En caso de que necesite descargar el cuaderno, haga clic [AQUÍ](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DS0105EN-SkillsNetwork/labs/Labs_V4/Getting_Started_with_JupyterNotebook-20221213-1670889600.jupyterlite.ipynb) o abra el laboratorio haciendo clic en este [enlace](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/jupyterlite/latest/lab/index.html?notebook_url=https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DS0105EN-SkillsNetwork/labs/Labs_V4/Getting_Started_with_JupyterNotebook-20221213-1670889600.jupyterlite.ipynb).

Jupiter Kernels

Bienvenido a «Jupyter Kernels». Después de ver este vídeo, podrá definir un núcleo y describir cómo trabajar con ellos. El núcleo de un cuaderno es un motor computacional que ejecuta el código contenido en un archivo de cuaderno. Existen núcleos de Jupyter para muchos lenguajes, y exploraremos algunos que son relevantes en la ciencia de datos. Cuando se abre un documento de Notebook, el núcleo relacionado se inicia automáticamente. Cuando se ejecuta el cuaderno, el núcleo realiza el cálculo y produce los resultados. Dependiendo de su configuración, es posible que necesite instalar otros lenguajes de cuadernos en su entorno de Jupyter. En el entorno de laboratorio de Skills Network, ya tiene preinstalados algunos idiomas. El primero es el núcleo de Python. Cuando lance un cuaderno, elija el idioma que le interese para su proyecto de ciencia de datos. El núcleo de Python le permite ejecutar celdas de Python. Puede ejecutar el script de Python en las celdas para generar una salida. En la esquina superior derecha del cuaderno se muestra el nombre del núcleo. Aquí se muestra el núcleo de Python. Tiene la opción de ejecutar otros núcleos. El entorno virtual Jupyter de Skills Network incluye Apache, Julia, R y Swift. Puede utilizar cualquier idioma para ejecutar el código, ya sea seleccionando el núcleo en la página de inicio o haciendo clic en el icono superior derecho y seleccionando el núcleo en el menú desplegable. Si ejecuta el núcleo en su máquina local, tendrá que instalar los idiomas manualmente a través de la interfaz de línea de comandos (CLI). En este vídeo, aprendió que el núcleo actúa como un motor computacional y ejecuta el código de un archivo de bloc de notas. Jupyter Notebook es compatible con distintos idiomas y puede cambiar a un núcleo diferente según sus necesidades.

**Laboratorio práctico: Uso de Markdown en cuadernos Jupyter**

**Estado: Traducido automáticamente del Inglés**

**Traducido automáticamente del Inglés**

Información:

Este elemento incluye contenido que aún no se tradujo a tu idioma preferido.

Los cuadernos Jupyter se componen de código y celdas markdown. Las celdas markdown se utilizan para presentar texto, imágenes, etc. y son útiles para documentar y explicar el contenido de su cuaderno. En este laboratorio, se familiarizará con la escritura de markdown.

Tras completar este laboratorio, será capaz de:

* Identificar en qué Kernel se está ejecutando su cuaderno Jupyter
* Crear encabezados y añadir texto en negrita y cursiva en Markdown
* Insertar enlaces e imágenes en Markdown
* Crear tablas en Markdown
* Crear listas ordenadas y desordenadas en Markdown
* Demostrar su destreza en el uso de Markdown en cuadernos Jupyter

Skills Network Labs (SN Labs) es un entorno de laboratorio virtual utilizado en este curso. Al hacer clic en**"Launch App**", de acuerdo con la política de privacidad de IBM Skills Network, su nombre de usuario y su dirección de correo electrónico se transmitirán a Skills Network Labs y sólo se utilizarán para comunicarle información importante para mejorar su experiencia de aprendizaje.

En caso de que necesite descargar el cuaderno, haga clic [AQUÍ](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DS0105EN-SkillsNetwork/labs/Labs_V4/Using_markdowns-20221213-1670889600.jupyterlite.ipynb) o abra el laboratorio haciendo clic en este [enlace](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/jupyterlite/latest/lab/index.html?notebook_url=https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DS0105EN-SkillsNetwork/labs/Labs_V4/Using_markdowns-20221213-1670889600.jupyterlite.ipynb).

**Jupyter Architecture».**

Bienvenido a «Jupyter Architecture». Tras ver este vídeo, podrá describir la arquitectura básica de Jupyter y explicar la arquitectura de Jupyter para la conversión de un formato de archivo. La arquitectura de Jupyter implementa un modelo de dos procesos con un núcleo y un cliente. El cliente es la interfaz que ofrece al usuario la posibilidad de enviar código al núcleo. Es el navegador de un Jupyter Notebook. El núcleo ejecuta el código y devuelve el resultado al cliente para que lo muestre. Los cuadernos de Jupyter representan el código, los metadatos, el contenido y los resultados. Al guardar el cuaderno, se envía desde el navegador al servidor de Notebook. Guarda el archivo del bloc de notas en un disco como un archivo JSON con la extensión.ipynb (que se pronuncia punto i PI NB). El servidor de cuadernos es responsable de guardar y cargar los cuadernos. Y el núcleo ejecuta las celdas de código contenidas en el cuaderno cuando el usuario las ejecuta. La arquitectura Jupyter utiliza la herramienta de conversión NB para convertir archivos a otros formatos. Por ejemplo, si queremos convertir un archivo de bloc de notas en un archivo HTML, un preprocesador modifica primero el bloc de notas y, a continuación, un exportador convierte el bloc de notas al nuevo formato de archivo. Por último, un postprocesador trabajará en el archivo exportado para obtener el resultado final. Después de la conversión, al proporcionar la URL del archivo, se muestra el archivo HTML. En este vídeo, aprendiste que: Jupyter implementa un modelo de dos procesos con un núcleo y un cliente. El servidor de cuadernos es responsable de guardar y cargar los cuadernos. El núcleo ejecuta las celdas de código contenidas en el cuaderno. Y la arquitectura Jupyter utiliza la herramienta de conversión NB para convertir archivos a otros formatos.

**Laboratorio práctico: Trabajar con archivos en cuadernos Jupyter**

**Estado: Traducido automáticamente del Inglés**

**Traducido automáticamente del Inglés**

Información:

Este elemento incluye contenido que aún no se tradujo a tu idioma preferido.

Después de completar este laboratorio, usted será capaz de:

* Explorar diferentes opciones de ejecución
* Cambiar el nombre de su portátil
* Apagar su cuaderno
* Guardar y descargar el cuaderno
* Cargar la libreta descargada
* Trabaje con varios cuadernos

Skills Network Labs (SN Labs) es un entorno de laboratorio virtual utilizado en este curso. Al hacer clic en "Abrir herramienta", de acuerdo con la política de privacidad de IBM Skills Network, su nombre de usuario y su dirección de correo electrónico se transmitirán a Skills Network Labs y sólo se utilizarán para comunicarle información importante para mejorar su experiencia de aprendizaje.

En caso de que necesite descargar el cuaderno, haga clic [AQUÍ](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DS0105EN-SkillsNetwork/labs/labs/Labs_V4/Working_with_files-20221121-1668988800.jupyterlite.ipynb).