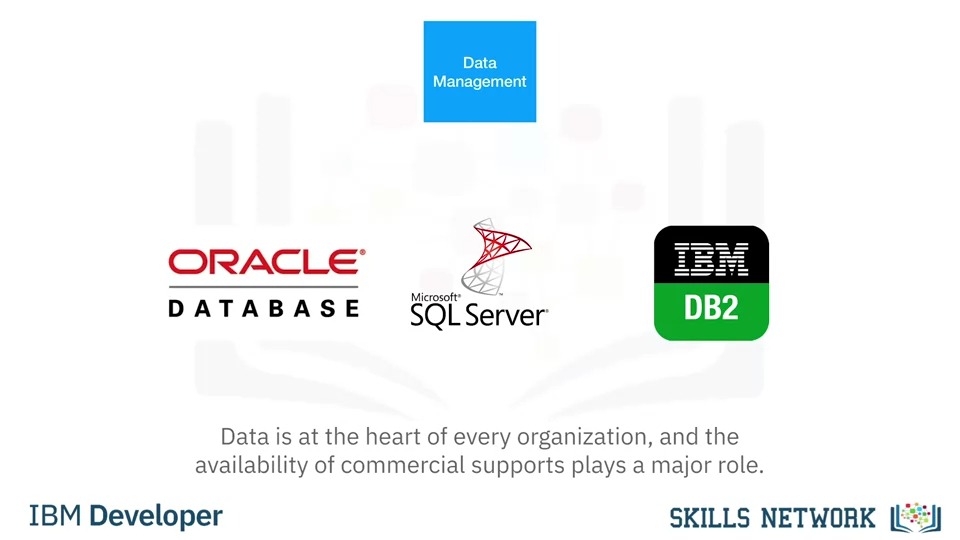
***Open source tools for data science:***

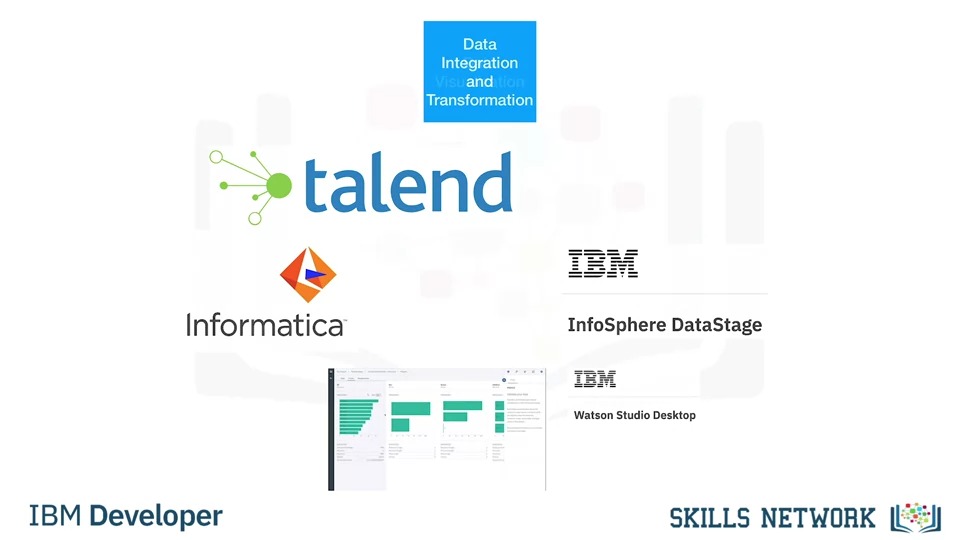
* **Data management:** the process of persisting and retrieving data.
* **Date integration and transformation:** referred as extract, transform, and load or “ETL”, is the process retrieving data from remote data management systems. Transforming data and loading it into a local management system is also part of data integration and transformation.
* **Data visualization:** is part of an initial data exploration process, as well as being part of a final deliverable.
* **Model building:** is the process of creating a machine learning or deep learning model using an appropriate algorithm with a lot of data.
* **Model deployment:** makes such a machine learning or deep learning model available to third-party applications.
* **Model monitoring and assessment:** ensures continuous performance quality checks on the deployed models. These checks are for accuracy, fairness, and adversarial robustness.
* **Code asset management:** uses versioning and other collaborative features to facilitate teamwork.
* **Data asset management:** brings the same versioning and collaborative components to data. Also supports replication, backup, and access right management.
* **Development environments:** commonly known as integrated development environments, or “IDEs”, are tools that help the data scientist to implement, execute, test, and deploy their work.
* **Execution environment:** are tools where data preprocessing, model training, and deployment take place.

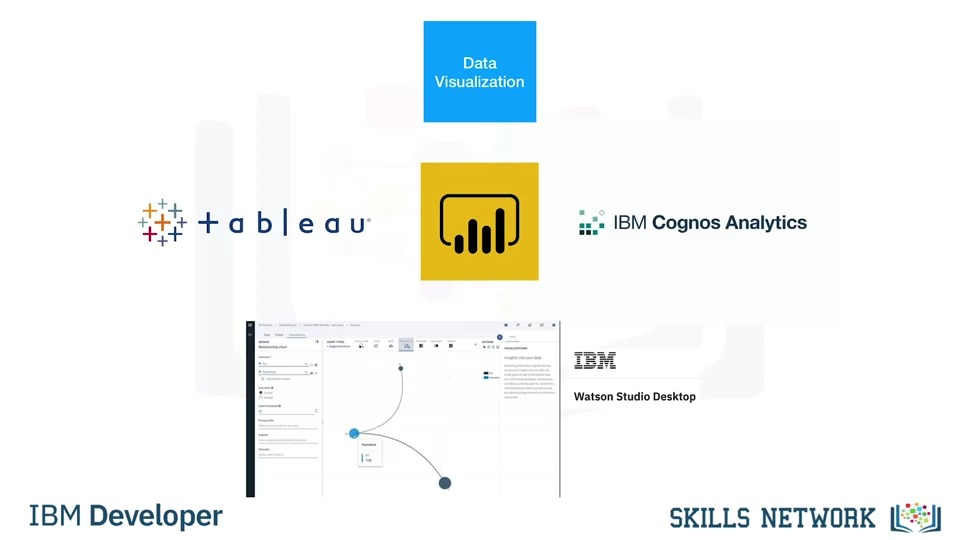
**Open source tools for data science**

* Data Management tools:
  + *Relational databases*: such as MySQL and PostgreSQL.
  + *NoSQL databases*: such as MongoDB, Apache CouchDB, and Apache Cassandra.
  + *File-based tools:* such as the Hadoop File system or Cloud File systems like Ceph.
  + *Elasticsearch*: is mainly used for storing text data and creating a search index for fast document retrieval.
* Data integration and transformation tools:
  + *Apache Airflow:* originally created by Airbnb.
  + *Kubeflow:* enables you to execute data science pipelines on top of Kubernetes.
  + *Apache Kafka:* originated from Linkedin.
  + *Apache Nifi:* delivers a very nice visual editor.
  + Apache SparkSQL: enables to use ANSI SQL and scales up to compute clusters of 1000s of nodes.
  + NodeRED: which also provides a visual editor, consumes so little in resources that it even runs on small devices like a Raspberry Pi.
* Data visualization tools:
  + Hue: can create visualizations from SQL queries.
  + Kibana: a data exploration and visualization web application, is limited to Elasticsearch (the data provider).
  + Apache Superset: is a data exploration and visualization web application.
* Model deployment tool:
  + Apache PredictionIO: currently only supports Apache Spark ML models for deployment, but support for all sorts of other libraries is on the roadmap.
  + Seldon: is an interesting product since it supports nearly every framework, including TensorFlow, Apache SparkML, R, and scikit-learn. Seldon can run on top of Kubernetes and Redhat OpenShift.
  + MLeap
  + Tensorflow: can serve any of its models using the TensorFLow service. You can deploy to an embedded device like a Raspberry Pi or a smartphone using **TensorFlow Lite**, and even deploy to a web browser using **TensorFlow dot JS.**
* Model monitoring and assessment tools:
  + *ModelDB:* a machine model metadatabase where information about the models are stored and can be queried. It natively supports Apache Spark ML Pipelines and scikit-learn.
  + *Prometheus:* a generic, multi-purpose tool, also widely used for machine learning model monitoring, although it’s not specifically made for this purpose.
  + *The IBM AI Fairness 360 bias:* detects and mitigates against bias in machine learning models.
  + *The IBM Adversarial Robustness 360 Toolbox*: can be used to detect vulnerability to adversarial attacks and help make the model more robust.
  + *The IBM AI Explainability 360 Toolkit*: makes the machine learning process more understandable by finding similar examples within a dataset that can be presented to a user for manual comparison. Can also illustrate input variables affect the final decision of the model.
* Code asset management tools (version management or version control):
  + *GitHub:* provides hosting for software development version management.
  + *GitLab:* has the advantage of being a fully open source platform that you can host and manage yourself.
  + *Bitbucket.*
* Data asset management tools (data governance or data lineage):
  + *Apache atlas:* a tool that supports that data has to be versioned and annotated with metadata.
  + *ODPi Egeria*: managed through the Linux Foundation and is an open ecosystem. It offers a set of open APIs, types, and interchange protocols that metadata repositories use to share and exchange data.
  + *Kylo:* an open source data lake management software platform that provides extensive support for a wide range of data asset management tasks.
* Development environment
  + *Jupyter:* first emerged as a tool for interactive Python programming; it now supports more than a hundred different programming languages through “kernels.”
    - Jupyter kernels are encapsulating the different interactive interpreters for the different programming languages.
    - A key property of Jupyter Notebooks is the ability to unify documentation, code, output from the code, shell commands, and visualizations into a single document.
  + *JupyterLab:* is the next generation of Jupyter Notebooks and in the long term, will actually replace Jupyter Notebooks. The architectural changes being introduced in JupyterLab makes Jupyter more modern and modular.
    - the ability to open different types of files, including Jupyter Notebooks, data, and terminals. You can then arrange these files on the canvas.
* *Apache Zeppelin*: it’s inspired by Jupyter Notebooks and provides a similar experience. One key differentiator is the integrated plotting capability.
* *RStudio:* unifies programming, execution, debugging, remote data access, data exploration, and visualization into a single tool.
* *Spyder:* tries to mimic the behaviour of RStudio to bring its functionality to the Python world. But in the Python world, Jupyter is used more frequently.
* Execution Environments
  + *Apache Spark:* The key property of Apache Spark is linear scalability. This means, if you double the number of servers in a cluster, you’ll also roughly double its performance.
    - Apache Spark is a batch data processing engine, capable of processing huge amounts of data file by file.
    - Is usually the choice in most use cases.
  + *Apache Flink:* is a stream processing image, with its main focus on processing real-time data streams.
  + *Ray:* has a clear focus on large-scale deep learning model training.
* Fully integrated visual tools: (most important tasks are supported by these tools, these tasks include data integration, transformation, data visualization, and model building)
  + *KNIME*: KNIME has a visual user interface with drag-and-drop capabilities. It also has built-in visualization capabilities.
    - Knime can be extended by programming in R and Python, and has connectors to Apache Spark.
* *Orange:* It’s less flexible than KNIME, but easier to use.

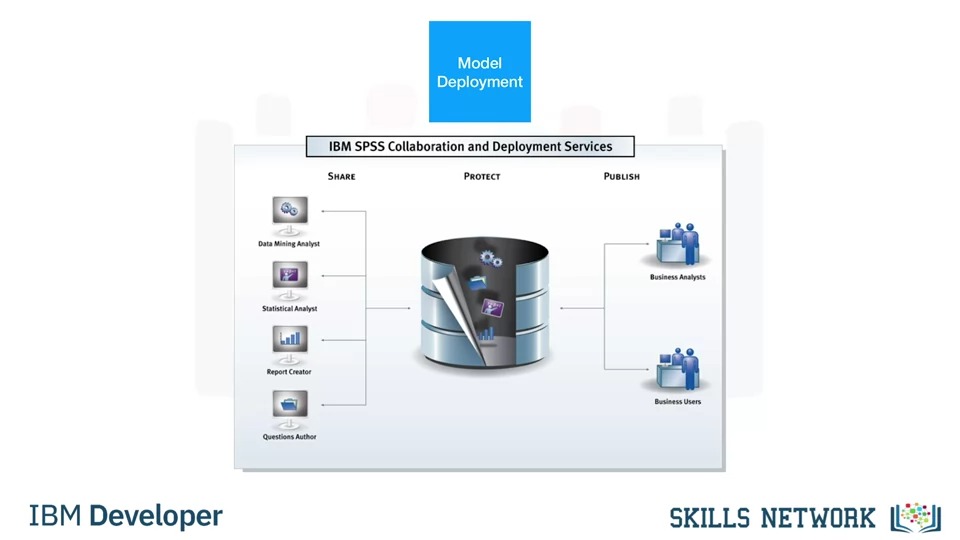
***Commercial tools for data science:***

******

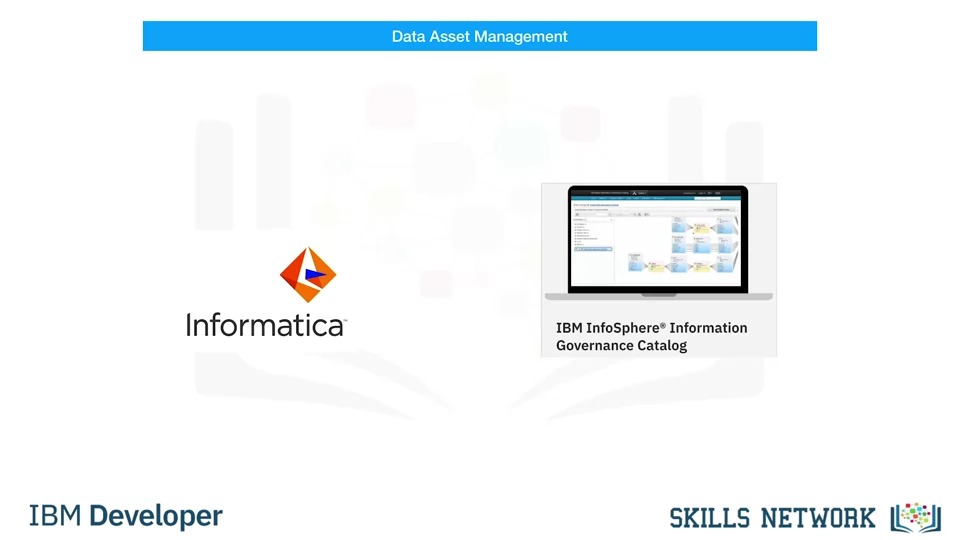
******

******

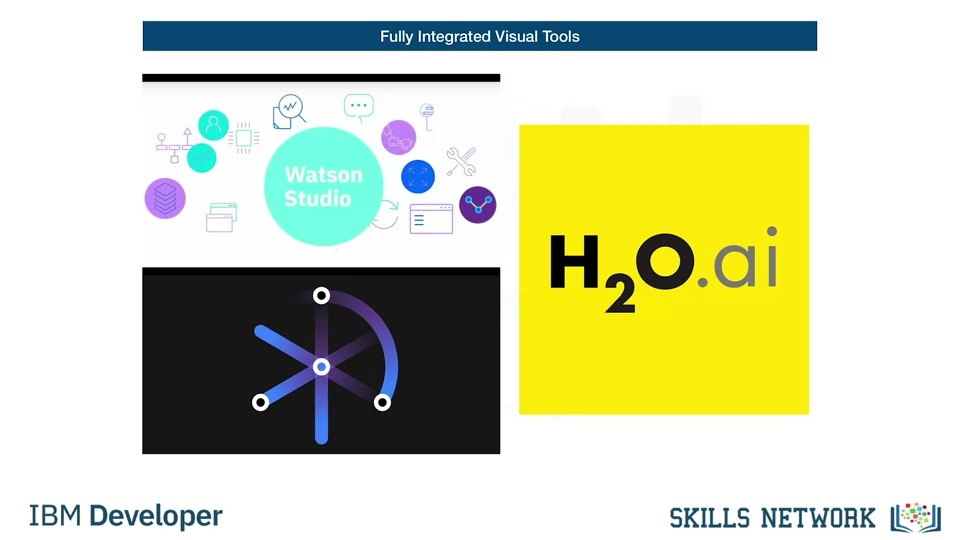
******

******

***There is no commercial tools for model monitoring and code asset management, open source is the first choice.***

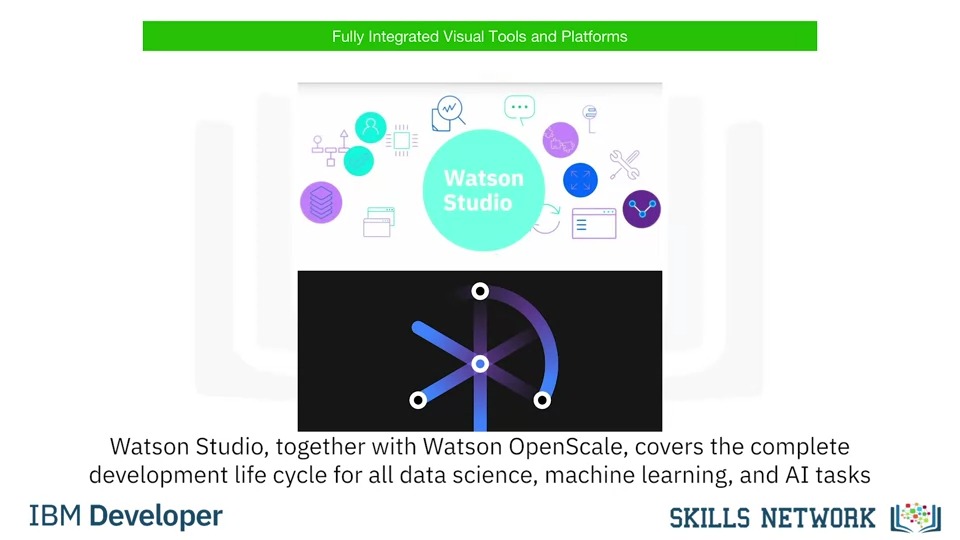
******

******

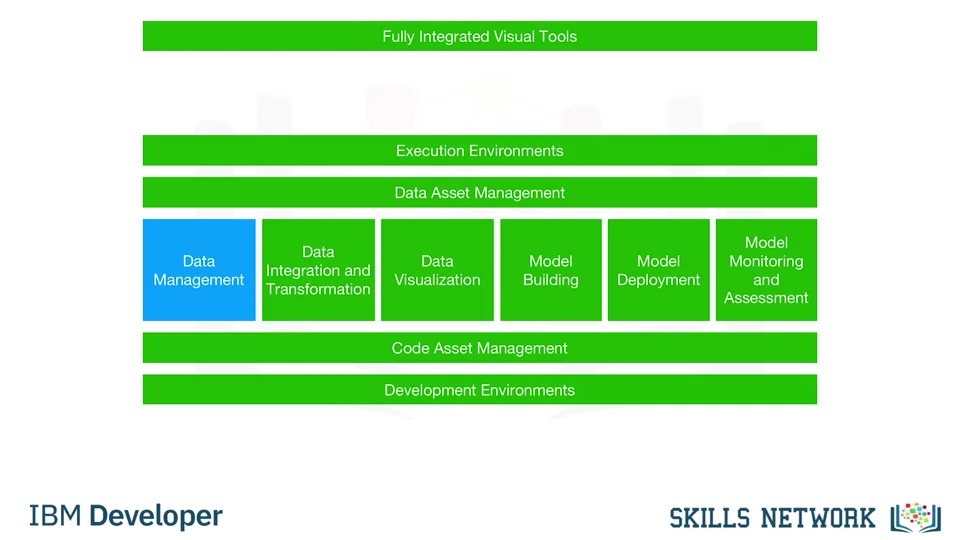
******

***Cloud based tools for data science***

* **Fully integrated visual tools and platforms:**
  + *Watson Studio and Watson OpenScale:*

******

* *Microsoft Azure Machine Learning:* This is also a fully cloud-hosted offering supporting the complete development life cycle of all data science, machine learning, and AI tasks.
* *H2O Driverless AI:* Although it is a product that you download and install, one-click deployment is available for the common cloud service providers.
* **Data Management:** 
  + *Amazon Web Services DynamoDB*: a NoSQL database that allows storage and retrieval of data in a key-value or a document store format.
  + *Cloudant:* which is a database-as-a-service offering. But, under the hood it is based on the open source *Apache CouchDB.*
    - It has an advantage: although complex operational tasks like updating, backup, restore, and scaling are done by the cloud provider, under the hood this offering is compatible with CouchDB.
    - Therefore, the application can be migrated to another CouchDB server without changing the application.
* *Db2:* This is an example of a commercial database made available as a software-as-a-service offering in the cloud, taking operational tasks away from the user.
* **Data integration tools:** (we talk not only about “extract, transform, and load,” or “ETL” tools, but also about “extract, load, and transform,” or “ELT,” tools. This means the transformation steps are not done by a data integration team but are pushed towards the domain of the data scientist or data engineer)
  + *Informatica Cloud Data integration*
  + *IBM’s Data Refinery:* Data Refinery enables transformation of large amounts of raw data into consumable, quality information in a spreadsheet-like user interface. Data Refinery is part of IBM Watson Studio.
* **Data visualization tools:** 
  + *Datameer*
  + *IBM Cognos Analytics*
  + *IBM data refinery*
* **Model building tools:** 
  + *Watson Machine Learning:* can train and build models using various open source libraries.
  + *Google cloud called AI Platform training*.
* **Model deployment tools:** 
  + *SPSS Collaboration and Deployment Services*: which can be used to deploy any type of asset created by the SPSS software tools suite.
  + *Watson Machine Learning:* can also be used to deploy a model and make it available to consumers using a REST interface.
* **Model monitoring and assessment tools:** 
  + *Amazon SageMaker Model Monitor:* continuously monitors deployed machine learning and deep learning models.



Everything marked in green can be using Watson Studio and Watson OpenScale.

***Libraries for Data Science***

Python libraries:

1. Scientific Computing Libraries in Python
   1. ***Pandas***: offers data structures and tools for effective data cleaning, manipulation, and analysis. It provides tools to work with different types of data. The primary instrument of Pandas is a two-dimensional table consisting of columns and rows.
   2. ***NumPy libraries***: are based on arrays, enabling you to apply mathematical functions to these arrays. Pandas is actually built on top of NumPy.
2. Visualization Libraries in Python (Data visualization methods are a great way to communicate with others and show the meaningful results of analysis)
   1. ***Matplotlib:*** is the most well-known library for data visualization, and it’s excellent for making graphs and plots. The graphs are also highly customizable.
   2. ***Seaborn:*** is based on matplotlib. Seaborn makes it easy to generate plots like heat maps, time series, and violin plots.
3. High level-Machine learning and deep learning
   1. ***Scikit-learn (machine learning):*** contains tools for statistical modeling, including regression, classification, clustering and others. It is built on NumPy, SciPy, and matplotlib, and it’s relatively simple to get started.
   2. ***Keras (deep learning)***: enables you to build the standard deep learning model. Like Scikit-learn, the high-level interface enables you to build models quickly and simply. It can function using graphics processing units (GPU), but for many deep learning cases a lower-level environment is required.
4. Deep learning libraries in Python
   1. ***TensorFlow:*** is a low-level framework used in large scale production of deep learning models. It’s designed for production but can be unwieldy for experimentation.
   2. ***Pytorch:*** is used for experimentation, making it simple for researchers to test their ideas.

**Libraries and other frameworks used in other languages**

* ***Apache Spark:*** is a general-purpose cluster-computing framework that enables you to process data using compute clusters. This means that you process data in parallel, using multiple computers simultaneously.
  + The Spark library has similar functionality as Pandas, Numpy and Scikit-learn.
  + Apache Spark data processing jobs can use Python R Scala, or SQL.
* ***Vegas****:* is a Scala library for statistical data visualizations. With Vegas, you can work with data files as well as Spark DataFrames.
* ***BigDL:*** for deep learning.
* ***R-***libraries:
  + Ggplot2: a popular library for data visualization in R.
  + Keras and TensorFlow.

**Application Programming Interfaces (API)**

What is API?

* An API lets two pieces of software talk to each other.

Rest APIs

* They enable you to communicate using the internet, taking advantage of storage, greater data access, artificial intelligence algorithms, and many other resources.
* In rest APIs, your program is called the “client”.
* The API communicates with a web service that you call through the internet.
* They have a set of Rules regarding:
  + Communication
  + Input or request
  + Output or response

**Data Sets - Powering Data Science**

* A data set is a structured collection of data. Data embodies information that might be represented as text, numbers, or media such as images, audio, or video files.
* Data structures
  + ***Tabular data:*** comprises a collection of rows, which in turn comprise columns that store the information. Ex: comma separated values or “CSV”. A CSV file is a delimited text file where each line represents a row and data values are separated by a comma.
  + ***Hierarchical or network data:*** are typically used to represent relationships between data. Hierarchical data is organized in a **tree-like structure,** whereas network data might be stored as a **graph**.
  + ***Raw files:*** such as images or audio. The MNIST dataset is popular for data science. It contains images of handwritten digits and is commonly used to train image processing systems.

**Data ownership**

* Private data
  + Confidential
  + Private or personal information
  + Commercially sensitive
* Open data (for free)
  + Scientific institutions
  + Governments
  + Organizations
  + Companies
  + Publicly available

**Open data** has played a significant role in the growth of data science, machine learning, and artificial intelligence and has provided a way for practitioners to hone their skills on a wide variety of data sets.

**Where to find open data**

* Open data portal list from around the world
* Governmental, intergovernmental and organization websites
* Kaggle
* Google data set search

**Community data license agreement**

* A Linux Foundation project
  + CDLA-sharing: permission to use and modify data; publication only under same terms
  + CDLA-permissive: permission to use and modify data; no obligations

**Sharing enterprise data – data asset exchange**

Data asset exchange (DAX):

* DAX provides a trusted source for finding open data sets that are ready for to use in enterprise applications.
* These data sets and which cover a wide variety of domains, including images, video, text, and audio.
* DAX data sets are typically easier to adopt, whether in research or commercial projects.
* DAX aims to make data sets available under one of the variants of the CDLACommunity Data License Agreement, in order to foster data sharing and collaboration.
* DAX also provides a single place to access unique data sets, in particular from IBM Research projects.
* DAX also provides tutorials in the form of notebooks that walk through the basics of data cleaning, pre-processing, and exploratory analysis.

**Machine learning models**

* Machine learning uses algorithms – also known as “models” - to identify patterns in the data.
* A model must be trained on data before it can be used to make predictions
* Machine learning models can be divided into three basic classes: supervised learning, unsupervised learning, and reinforcement learning.

**Supervised learning:**

* The model tries to identify relationships and dependencies between the input data and the correct output.
* Used to solve regression and classification problems.
  + Regression models are used to predict a numeric, or “real” value. Ex: home sales prices, stock market prices
  + Classification models are used to classify things into categories. Ex: email spam filters, fraud detection, image classification.

**Unsupervised learning:**

* Data is not labeled
* Model tries to identify patterns without external help
* Common learning problems: clustering and anomaly detection

**Reinforcement learning:**

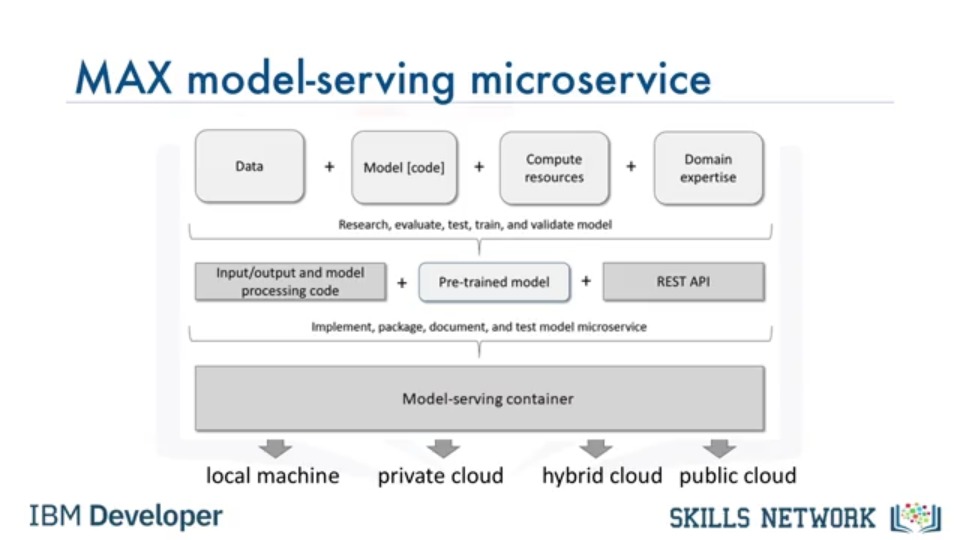
* Conceptually similar to human learning processes
* E.g. a robot learning to walk; chess, go and other games of skill

**Deep learning**

* Tries to loosely emulate how the human brain works
* Applications
  + Natural language processing
  + Image, audio, and video analysis
  + Time series forecasting
  + Much more
* Requires typically very large datasets of labeled data to train a model and is compute intensive and usually requires special purpose hardware to achieve acceptable training times.
* You can build a custom deep learning model from scratch or use pre-trained models from public model repositories.
* Built using frameworks, such as:
  + TensorFlow
  + PyTorch
  + Keras
* Popular model repositories
  + Most frameworks provide a “model zoo”
  + ONNX model zoo

**The Model Asset exchange (reduces time to value)**

* Free open-source deep learning microservices
  + Use pre-trained or custom-trainable state-of-the-art models
  + Fully tested, deploy in minutes
  + Approved for personal and commercial use
* Available for variety of domains:
  + Object detection (“which objects are in this image”)
  + Image, audio, and text classification (“what is in this…”)
  + Named entity recognition (“identify entities in text”)
  + Image-to-text translation (“generate image caption”)
  + Human pose detection



**Model-serving microservice API**

* Model-serving microservices expose standardized REST API

**What is a Jupyter Notebook?**

* Jupyter Notebook is a tool for recording Data Science experiments.
* It allows a Data Scientist to combine text and code block in a single file.
* It generates plots and tables within the file
* Notebooks can be exported as pdf and html files

**Jupyter lab**

* Jupyter Lab is a browser-based application that allows you to access multiple Jupyter Notebook files as well as other code and data files.
* Jupyter Lab extends the functionalities of Jupyter notebooks by enabling you to work with multiple notebooks, text editors, terminals, and custom components in a flexible, integrated, and extensible manner.
* JupyterLab allows for: Interactive control of the notebook cells and output, real time editing markdowns, CSV etc. It is compatible with several file formats like CSV, JSON, PDF, Vega and more. And is open source.
* Jupyter notebooks can be used with a cloud-based service like IBM and Google Collab.

**Jupyter Kernels**

* A notebook kernel is a computational engine that executes the code contained in a Notebook file
* Jupyter Kernels for many other languages exist
* When the notebook is executed, the kernel performs the computation and produces the results.

**Jupyter Architecture**

* Jupyter implements a two-process model, with a kernel and a client
* The client is the interface offering the user the ability to send code to the kernel
* The kernel executes the code and returns the result to the client for display
* The client is the browser when using a Jupyter notebook
* Jupyter notebooks is used to represent code, metadata, contents and outputs
* It has a flexible interface that extends beyond code and output