**CATEGORIES OF DATA SCIENCE TOOLS**

Welcome to Categories of Data Science Tools. After watching this video, you will be able to list the tasks that a data scientist needs to perform, show how code asset management and data asset management help build models, and describe how execution and development `environments implement a model. Before it can be useful, raw data must pass through various

Data Science task categories, such as data management, data integration and transformation, data visualization, model building, model deployment, and model monitoring and assessment. To do these tasks, you need data asset management, code asset management, execution environments, and development environments. Let’s see how each category enables you to make the best use of raw data.

**Data Management** is the process of collecting, persisting, and retrieving data securely, efficiently, and cost-effectively. Data is collected from many sources, like Twitter, Flipkart, Media, Sensors, and more. Store collected data in persistent storage so it is available whenever you need it.

**Data Integration and Transformation**, is the process of **E**xtracting, **T**ransforming, and **L**oading data. This is called “**ETL**”. Some of this data is distributed in multiple repositories. For example, a database, a data cube, and flat files. Use the Extraction process to extract data from these numerous repositories and save to a central repository like a Data Warehouse. Data Warehouses are primarily used to collect and store massive amounts of data for data analysis. Next, **Data Transformation** is the process of transforming the values, structure, and format of data.

After extracting the data, the next step is to transform the data. In this example, height and weight data needs to be transformed to metric. And once the data is transformed, it’s time to load the data. Transformed data is loaded back to the Data Warehouse.

**Data visualization** is the graphical representation of data and information. You can use visualization to represent data in the form of charts, plots, maps, animations, etc. And data visualization conveys data more effectively for decision-makers. It is a crucial step in the data science process. Various forms of data visualizations include a bar chart, which compares the size of each component, a tree map, which displays hierarchy data, a line chart, which plots a series of data points over time, and a map chart, which displays data by location. Map charts can also be applied to other locations like websites. Now,

**Model Building** is the next step. This is where you train the data and analyze patterns with machine learning algorithms. The system ‘learns’ how to provide predictions or decisions by itself. You can then use this model to make predictions on new, unseen data. Model building can be done using a service called IBM Watson Machine Learning. It provides a full range of tools and services for building models. The next step is

**Model Deployment**: the process of integrating a developed model into a production environment. In model deployment, a machine learning model is made available to third-party applications via APIs (Application Programing Interface). Business users can access and interact with the data through these third-party applications. And this helps them make data-based decisions. As an example, the SPSS Collaboration and Deployment Services can be used to deploy any type of asset created by the SPSS software tools suite.

**Model Monitoring And Assessment** run continuous quality checks to ensure a model’s accuracy, fairness, and robustness. Model monitoring uses tools like Fiddler to track the performance of deployed models in a production environment. Now,

**Model Assessment** uses evaluation metrics like the F1 score, true positive rate, or the sum of squared error to understand a model's performance. A well-known example is the IBM Watson Open scale, which continuously monitors deployed machine learning and deep learning models. It will improve the accuracy and quality of your predictions. So, now that we’ve reviewed the data science task categories, let’s look at some of the tools that support them.

**Tools That Support The Data Science Task Categories**

**Code asset management** provides a unified view where you manage an inventory of assets. When you want to develop a model, you may need to update it, fix bugs, or improve code features incrementally. All of this requires version control. Developers use versioning to track and manage changes to a software project’s code. When working on a model, teams a centralized repository where everyone can upload, edit, and manage the code files simultaneously. Collaboration allows diverse people to share and update the same project together. GitHub is a good example of a code asset management platform. It’s web-based and provides sharing, collaboration, and access control features. As a data scientist, you want to properly store and organize all your images, videos, text, and other data in a central location. You also want control over who can access, edit, and manage your data.

**Data asset management**, also called **digital asset management (DAM),** is the organizing and managing of important data collected from different sources. DAM is performed on a DAM platform that allows versioning and collaboration. DAM platforms also support replication, backup, and access right management for the stored data.

**Development Environments**, also called

Integrated Development Environments, or “IDEs”, provide a workspace and tools to develop, implement, execute, test, and deploy source code. IDEs like IBM Watson Studio provide testing and simulation tools to emulate the real world so you can see how your code will behave after it is deployed.

**An execution** **environment** has libraries to compile the source code and system resources that execute and verify the code. Cloud-based execution environments are not tied to any specific hardware or software, and offer tools like IBM Watson Studio for data preprocessing, model training, and deployment. Finally, fully integrated visual tools like IBM Watson Studio and IBM Cognos Dashboard Embedded cover all the previous tooling components, and can be used to develop deep learning and machine learning models. In this video, you learned that the Data Science Task Categories are: Data Management, Data Integration and Transformation, Data Visualization, Model Building, Model Deployment, and Model Monitoring and Assessment. Data Science Tasks are supported by Data Asset Management, Code Asset Management, Execution Environments, and Development Environments.

**OPEN-SOURCE TOOLS**

For Data Science Part 1: After watching this video, you will be able to: List the open-source data management tools.

* List the open-source data integration and transformation tools.
* List the data visualization tools.
* List the model tools for building, deployment, monitoring, and assessment, and
* List tools for code and data asset management.

So, **the most widely used open-source data management tools** are:

**relational databases** like

* MySQL and PostgreSQL. Also, there are

**NoSQL Databases** like

* MongoDB,

Apache CouchDB, and Apache Cassandra. In addition, there are **file-based tools like** the Hadoop File System or

Cloud File systems like Ceph.

You also have an elastic **search tool** that stores text data, including the creation of a search index fo r fast document retrieval. Now, the task of data integration and transformation in the classic data warehousing world is to

Extract, Transform, and Load (ETL). Data scientists often propose Extract, Load,

Transform (ELT) as data is dumped somewhere, and the data engineer or data scientist handles the transformation of the data. Another term for this process emerged: Data Refinery and Cleansing. The most widely used open-source **data** **integration and transformation tools** are the following: Apache AirFlow, which was created by Airbnb originally. KubeFlow, which allows the execution of data science pipelines on top of Kubernetes. Apache Kafka, which originated from LinkedIn. Apache Nifi, which delivers a very nice visual editor. Apache SparkSQL, lets you use ANSI SQL and scales up to compute clusters of thousands of nodes and NodeRED also brings a visual editor. In addition, NodeRED is so low in resource consumption that it even runs on tiny devices like a Raspberry Pi. Now let’s discuss the most widely used open-source data visualization tools. You must distinguish between programming libraries where you must use code or tools containing a user interface. Pixie Dust is also a library but has a user interface that facilitates plotting in Python. A similar approach uses Hue, which can create visualizations from SQL queries. Whereas Kibana, a data exploration, and visualization web application is limited to Elasticsearch (data provider). And finally, Apache Superset is a data exploration and visualization web application.

**Model deployment** is a crucial step. Once you’ve created a machine learning model capable of predicting some critical aspects of the future, you should make it consumable

by other developers and turn it into an API. Apache PredictionIO currently only supports

Apache Spark ML models for deployment, but support for all libraries is on the roadmap. Seldon is an interesting product since it supports nearly every framework including, TensorFlow, Apache SparkML, R, and scikit learn. Interestingly, it can run on top of Kubernetes and Redhat OpenShift. Another way to deploy SparkML models is MLeap. Finally, TensorFlow can serve any tensor flowmodel using the TensorFlow service. It can be an embedded device like

a Raspberry Pi or smartphone using TensorFlow lite and deployed to a

web browser using TensorFlow dot JS.

**Model monitoring** is an important step as well. Once you’ve deployed a machine learning model, you want to track its prediction performance while

new data arrives to maintain outdated models. Some examples are the following: ModelDB is a machine model metadata base where information about the

models is stored and queried. It natively supports Apache Spark

ML Pipelines and scikit-learn. A generic, multi-purpose tool called

Prometheus is widely used as well. Although it is not specifically made

for machine learning model monitoring, it is used for this purpose. Model performance is measured

by more than accuracy. Model bias against protected groups like

gender or race is important as well. The IBM AI Fairness 360 open-source toolkit detects and mitigates bias

in machine learning models. These models, especially neural network-based deep

learning models, can be subject to adversarial attacks where an attacker tries to mislead the

model with manipulated data or by controlling it. The IBM Adversarial Robustness 360 Toolbox detects vulnerability against adversarial attacks

and leverages the model to be more robust. Finally, machine learning modes are often

considered as a black box applying some magic. The IBM AI Explainability 360 toolkit

addresses that problem by finding similar examples in a dataset to be presented

to an end-user for manual comparison. IBM AI Explainability 360 toolkit can

also address the training of a simpler machine learning model to explain

the responsibility of different input variables directed toward

the final decision of the model. So, the choice of code asset management

tools has become quite simple. Git is now the de facto standard

for code asset management, also known as version management or version control. Around Git emerged several services. The most prominent is GitHub, but the

runner-up is GitLab, with the advantage that the platform is entirely open source

and can be hosted and managed on your own. Another choice is Bitbucket. Data asset management, also known as

data governance or data lineage, is a crucial part of enterprise-grade data science. Data has to be versioned

and annotated with metadata. Apache Atlas is such a tool supporting this task. Another interesting project is ODPi Egeria, managed through the Linux Foundation, is an

open ecosystem offers a set of open APIs, types, and interchange protocols that metadata

repositories use to share and exchange data. And finally, Kylo is an open-source

data management software platform, with extensive support for

data asset management tasks. In this video, you learned that:

**Data management** tools are

* MySQL,
* PostgreSQL,
* MongoDB,
* Apache CouchDB,
* Apache Cassandra, Hadoop File System,
* Ceph, and
* elastic search.

**Data integration and transformation tools** are

* Apache AirFlow, -
* KubeFlow,
* Apache Kafka,
* Apache Nifi,
* Apache SparkSQL, and
* NodeRED.

**Data Visualization tools are**

* Pixie Dust,
* Hue,
* Kibana, and
* Apache Superset.

**Model Development Tools are**

* **IBM Watson Studio:** Engineered as an integrated environment, Watson Studio simplifies developing, training, and deploying models. It boasts support for multiple languages and frameworks, such as Python, R, and TensorFlow, alongside collaboration features, data preparation tools, and versatile deployment options.
* **IBM AutoAI:** A notable feature embedded within Watson Studio, IBM AutoAI streamlines the machine learning model construction process. By dynamically exploring various algorithms and hyperparameters, it aims to identify the optimal model for a given dataset.
* **IBM Watson OpenScale:** As a platform for overseeing and managing AI models in production, Watson OpenScale plays a pivotal role in ensuring model fairness, explainability, and bias mitigation. It furnishes insights into model performance and drift over time, facilitating informed decision-making.
* **IBM Watson Machine Learning:** Watson Machine Learning, available as a service on the IBM Cloud platform, enables users to scale their training and deployment of machine learning models. It seamlessly supports popular frameworks like TensorFlow, PyTorch, and scikit-learn, and offers APIs for seamless integration with other applications.

**Model deployment tools** are

* Apache PredictionIO,
* Seldon,
* Kubernetes,
* Redhat OpenShift,
* Mleap,
* TensorFlow service,
* TensorFlow lite, and
* TensorFlow dot JS.

**Model monitoring tools** are

* ModelDB,
* Prometheus,
* IBM AI Fairness 360,
* IBM Adversarial Robustness 360 Toolbox, and
* IBM AI Explainability 360.

**Code asset management tools** are

* Git,
* GitHub,
* GitLab, and
* Bitbucket. And finally,

**Data Asset Management Tools**

are

* Apache Atlas,
* ODPi Egeria, and
* Kylo.

**Open-Source Tools for Data Science Part 2**

After watching this video, you will be able to Compare and contrast different open-source tools, and describe the relevant features of open-source tools. Currently, the most famous development environment data scientists are using is “Jupyter,” which emerged as a tool for interactive Python programming. Jupyter now supports more than a hundred different programming languages through “kernels.” This encapsulates the execution environment for the different programming languages. A key property of Jupyter Notebooks

is to unify documentation, code, output from the code, shell commands, and visualizations in a single document. Jupyter lab is the next version of Jupyter Notebooks, and in the long term will replace Jupyter Notebooks. The abundance of architectural changes makes Jupyter more modern and modular. From a user’s perspective, the main

difference between Jupyter Lab and Jupyter Notebooks is the ability

to open different types of files, including Jupyter Notebooks, data, and

terminals, and then arrange them on the canvas. Although it has been reimplemented from scratch, Apache Zeppelin was inspired by Jupyter

Notebooks and provides a similar experience. One key differentiator is the

integrated plotting capability. In Jupyter Notebooks, you are required

to use external libraries in Zeppelin, and plotting doesn’t require coding. You can also extend the capabilities

by using additional libraries. RStudio is among the oldest development

environments for statistics and data science. RStudio has its origins in the year 2011. It exclusively runs R and

all associated R libraries. In the R environment, Python

development is possible. R is tightly integrated into the Jupyter

tool and provides optimal user experience. RStudio unifies programming, execution, debugging, remote data access, data exploration, and visualization into one tool. Finally, Spyder tries to mimic

the behavior of RStudio to bring its functionality to the Python world. Although not at par with the

functionality of RStudio, data scientists consider it as an alternative.

In the Python world, Jupyter is used more. This diagram shows that Spyder integrates code, documentation, and visualizations,

among others, into a single canvas. Sometimes your data doesn’t fit into a single

computer’s storage or main memory capacity. Therefore, cluster execution environments exist. The extensively famous Apache Spark is

among the most active Apache projects that are used across all industries,

including many Fortune 500 companies. The key property of Apache

Spark is linear scalability. This means that if you double the

number of servers in a cluster, you’ll roughly double its performance. Apache Flink was developed after Apache

Spark continued to gain market share. The key difference between Apache Spark and

Apache Flink is that Apache Spark is a batch data processing engine, capable of processing

vast amounts of data one by one or file by file. Whereas Apache Flink is a stream-processing image with its main focus on processing

real-time data streams. Although both engines support both data

processing paradigms, at the same time, Apache Spark is the choice for most use cases. After Apache Spark and Apache Flink, Ray is one of the latest developments in

the data science execution environments and has a clear focus on large-scale

deep learning model training. Let’s look at open-source tools for data

scientists, which are fully integrated and visual. This means no programming knowledge is necessary. The tools support a subset of

important tasks that include data integration and transformation, data visualization, and model building.

**KNIME** originated from the

University of Konstanz in 2004. As you can see, KNIME has a visual user

interface with drag-and-drop capabilities. It has built-in visualization

capabilities. In addition, it can be extended by programming in R and

Python and even has connectors to Apache Spark.

**Orange** is another representative of this group of tools. It is less flexible than KNIME but is easier to use. In this video, you have learned about the most common tasks in data science and which open-source tools are relevant.

**COMMERCIAL TOOLS FOR DATA SCIENCE**

After watching this video, you will be able

* List the commercial data management tools,
* List the commercial data integration and transformation tools
* List the data visualization tools
* List the model tools for building, deployment, monitoring, and assessment, and
* List tools for code asset management, data asset management, development environment, and fully integrated visual.

Let’s revisit the overview of different tool categories.

In **Data Management**, most of an enterprise’s relevant data is stored in an **Oracle Database**, in **Microsoft SQL Server**, or an **IBM Db2**. Although open-source databases are coming to the forefront, these three data management products are considered industry-standard and will be around for a while. In addition, it’s not only about functionality. Since data is the heart of every organization, commercial support availability plays a major role. Commercial supports are delivered directly from software vendors, influential partners, and support networks.

Let’s start with **Commercial Data Integration Tools** that comprise extract, transform, and load (ETL) tools. According to a Gartner Magic Quadrant, Informatica PowerCenter and IBM InfoSphere DataStage are the leaders. These are followed by SAP, Oracle, SAS, Talend, and Microsoft products. These tools support the design and deployment of ETL data processing pipelines through a graphical interface. They bring along connectors to most of the commercial and open-source target information systems. Finally, Watson Studio Desktop includes a component called Data Refinery, which enables definition and execution of data integration processes in a spreadsheet-style. In the commercial environment, **DATA VISUALIZATIONS** use business intelligence (BI) tools. The focus of these tools is to create visual reports and live dashboards. The most prominent commercial representatives are: Tableau, Microsoft Power BI, and IBM Cognos Analytics. Another type of visualization targets data scientists rather than end users. For example, the visualization can show relationships between different columns in a table. This functionality is contained in Watson Studio Desktop.

If you want to build a **Machine Learning Model** with a commercial tool, you should use a data mining product. The most prominent products in that space are: SPSS Modeler and SAS enterprise miner. In addition, SPSS Modeler is also available in Watson Studio Desktop, based on the tool’s cloud version.

Now, **MODEL DEPLOYMENT** in commercial software is tightly integrated into the model-building process. Here is an example of the PSSS Collaboration and Deployment Services, which is used to deploy any type of asset created by the SPSS software tools suite. The same holds for other vendors. Also, commercial software can export models in an open format. For example, SPSS Modeler supports exporting models as predictive model markup language (PMML), which an abundance of other commercial and open software packages can read. Model monitoring is a very new discipline. Currently, relevant commercial tools are not available. Therefore, open source is the first choice. The same is true for code asset management. Open source with Git and GitHub is the de facto standard. Data asset management, often called data governance or data lineage, is a crucial part of enterprise-grade data science. Data must be versioned and annotated with metadata. Vendors, including Informatica Enterprise Data Governance and IBM, provides tools for these specific tasks. The Information Governance Catalog covers functions like a data dictionary, which facilitates the discovery of data assets. Each data asset is assigned to a data steward

or the data owner. The data owner is responsible for that data

asset and can be contacted. Then, data lineage is covered, allowing tracking

back the transformation steps in creating the data assets. The data lineage also includes a reference

to the actual source data. Rules and policies can be added to reflect

complex regulatory and business requirements for data privacy and retention. Watson Studio is a fully integrated development

environment for data scientists. Most people consume it through the cloud. And there is also a desktop version available. Watson Studio Desktop combines Jupyter Notebooks

with graphical tools to maximize the performance of data scientists. Watson Studio, together with Watson Open Scale,

is a fully integrated tool covering the data science life cycle involving all tasks discussed

previously. They can be deployed in a local data center,

on top of Kubernetes / RedHat OpenShift. Another example of a fully integrated commercial

tool is H2O Driverless AI, which covers the complete

data science life cycle. In this video, you learned that:

Commercial tools support the most common tasks in data science. Data management tools are Oracle Database,

Microsoft SQL Server, and IBM Db2. Data integration tools are mainly provided by

Informatica PowerCenter and IBM InfoSphere DataStage. These are followed by products from SAP, Oracle,

SAS, Talend, Microsoft, and Watson Studio Desktop. Model building tools are SPSS Modeler, and

SAS enterprise miner. SPSS Modeler is also available in Watson Studio

Desktop. Informatica and IBM provide data asset management

tools. And finally, Watson Studio, together with Watson Open Scale is a fully integrated tool covering the data science life cycle.

**CLOUD BASED TOOLS FOR DATA SCIENCE**

After watching this video, you will be able

to: Describe how commercial cloud tools support

data science tasks, and Explain how integration provides the ability

to use the same tools for multiple tasks. Let’s again look at the overview of different

tool categories. Since cloud products are a newer species,

they follow the trend of having multiple tasks integrated in tools. This integration is applicable for the tasks

marked green in the diagram. Let’s start with the fully integrated visual

tools category. Since these tools introduce a component where

large-scale execution of data science workflows happens in compute clusters, we have changed

the title and added the word “Platform.” These clusters are composed of multiple server machines, transparently for

the user, in the background. Watson Studio and Watson OpenScale cover the

complete development life cycle for all data science, machine learning, and artificial

intelligence (AI) tasks. Another example is Microsoft Azure Machine

Learning. It is also a full cloud-hosted offering supporting

the complete development life cycle of all data science, machine learning, and AI tasks. And finally, another example is H2O Driverless

AI. Although it is a product you download and

install, there exists a one-click deployment for the standard cloud service providers. Since the cloud provider does not do operations

and maintenance, as with Watson Studio, Open Scale, and Azure Machine Learning, this delivery

model should be distinct from Platform or Software as a service - PaaS or SaaS. In data management, with some exceptions,

software-as-a-service (SaaS) versions of existing open source and commercial tools exist. The cloud provider operates the tool for you

in the cloud. For example, the cloud provider operates the

product by backing up your data and configuring and installing updates. Some proprietary tools are only available

from a single cloud provider. One example of such a service is Amazon Web

Services DynamoDB, which is a NoSQL database. It allows storage and retrieving data in a

key-value or a document store format. The most prominent document data structure

is JSON. Another flavor of such a service is Cloudant,

which is a database as a service offering. But, in the background, it is based on the

open-source Apache CouchDB. The advantage is that complex operational

tasks like updating, backup, restoring, and scaling are done by the cloud provider. However, the Cloudant service offering is

compatible with CouchDB. Therefore, the application migrates to another

CouchDB server without making any changes to the application. IBM offers Db2 as a service as well. It is an example of a commercial database

made available as a SaaS offering in the cloud, taking away operational tasks from the user. Now let’s discuss commercial data integration

tools that include extract, transform, and load (ETL) tools and extract, load, and transform

(ELT) tools. It means the transformation steps are not

done by a data integration team but are pushed toward the domain of the data scientist or

data engineer. Two commercial data integration tools widely

used are Informatica Cloud Data Integration and IBM’s Data Refinery. Data Refinery is part of IBM Watson Studio. It allows transforming large amounts of raw

data into consumable, quality information in a spreadsheet-like user interface. So, the market for cloud data visualization tools

is huge, and every major cloud vendor has one. An example of a smaller company offering a

cloud-based data visualization tool is Datameer. IBM offers its famous Cognos Business intelligence

suite as a cloud solution. IBM Data Refinery also offers data exploration

and visualization functionality in Watson Studio. Again, those are examples of a rapidly changing

and growing commercial ecosystem among many established, and emerging vendors. In Watson Studio, various visualizations depict

data for better understanding. An example is this 3D bar chart that visualizes

a target value on the vertical dimension that is dependent on two other values in the horizontal

dimensions. You can use colors to visualize the third

dimension. Another data visualization is hierarchical

edge bundling that depicts correlations and affiliations between entities. If sufficient, a classic bar chart can do

the job as well. A 2D scatter plot with a heat map shows two

dependent data fields on the y-axis with different color intensities. A tree map shows the distribution of subsets

within a set. The famous pie chart does the same but in

a non-hierarchical manner. Finally, a word cloud pops out significant

terms in a document corpus. Model building can be done using a service. One example of a service is Watson Machine

Learning. Watson Machine Learning can train and build

models using various open-source libraries. Google has a similar service on their cloud

called AI Platform Training. Every cloud provider has a solution for this

task. Model deployment in commercial

software is usually tightly integrated into the model-building process. Here is an example of the SPSS Collaboration and Deployment Services which can be used to deploy any asset created by the SPSS software tools suite. The same holds for other vendors. In addition, commercial software

can export models in an open format. For example, SPSS Modeler supports exporting

models as Predictive Model Markup Language (PMML), which other commercial and

open software packages can read. In addition, Watson Machine Learning deploys a model and makes it available to

consumers using a REST interface. Amazon SageMaker Model Monitor is an example

of a cloud tool to monitor deployed machine learning and deep learning models continuously. Every major cloud provider has similar tooling. Another tool for model

monitoring is Watson OpenScale. Everything marked in green can be done

using Watson Studio and OpenScale. In this video, you’ve learned: Watson Studio and Watson OpenScale, cover the complete development life cycle for

all data science, machine learning, and AI tasks. In data management, with some exceptions, there

exists a software-as-a-service (SaaS) version of existing open-source and commercial tools. Two commercial data integration tools widely used are Informatica Cloud Data Integration and IBM’s Data Refinery. An example of a cloud-based data

visualization tool is Datameer and IBM’s Cognos Business intelligence suite. Model building can be done using a service such as Watson Machine Learning. Amazon SageMaker Model Monitor is an example of a cloud tool to monitor

deployed machine-learning and deep learning models continuously.