Open source tools are available for various data science tasks.

In this video, we’ll have a look at the different data science tasks.

In subsequent videos we’ll walk through the most commonly used open source tools for

those tasks.

The most important tools are covered throughout this course.

Data Management is the process of persisting and retrieving data.

Data Integration and Transformation, often referred to as Extract, Transform, and Load,

or “ETL,” is the process of retrieving data from remote data management systems.

Transforming data and loading it into a local data 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.

Data asset management 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 environments are tools where data preprocessing, model training, and deployment

take place.

Finally, there is fully integrated, visual tooling available that covers all the previous

tooling components, either partially or completely.

This concludes this video.

In the next video we’ll start looking at open source tools for data science tasks.

In part one of this two-part series, we’ll cover data management, open source data integration,

transformation, and visualization tools.

The most widely used open source data management tools are relational databases such as

MySQL and PostgreSQL; NoSQL databases such as MongoDB Apache CouchDB, and Apache Cassandra;

and file-based tools such as the Hadoop File System or Cloud File systems like Ceph.

Finally,Elasticsearch is mainly used for storing text data and creating a search index for

fast document retrieval.

The task of data integration and transformation in the classic data warehousing world is called

ETL, which stands for “extract, transform, and load.”

These days, data scientists often propose the term “ELT” – Extract, Load, Transform“ELT”,

stressing the fact that data is dumped somewhere and the data engineer or data scientist themself

is responsible for data.

Another term for this process has now emerged: “data refinery and cleansing.”

Here are the most widely used open source data integration and transformation tools:

Apache AirFlow, originally created by AirBNB; KubeFlow, which enables you to execute data

science pipelines on top of Kubernetes; Apache Kafka, which originated from LinkedIn;

Apache Nifi, which delivers a very nice visual editor;

Apache SparkSQL (which enables you to use ANSI SQL and scales up to compute clusters

of 1000s of nodes), and NodeRED, which also provides a visual editor.

NodeRED consumes so little in resources that it even runs on small devices like a Raspberry

Pi.

We’ll now introduce the most widely used open source data visualization tools.

We have to distinguish between programming libraries where you need to use code and tools

that contain a user interface.

The most popular libraries are covered in the next videos.

A similar approach uses Hue, which can create visualizations from SQL queries.

Kibana, a data exploration and visualization web application, is limited to Elasticsearch

(the data provider).

Finally, Apache Superset is a data exploration and visualization web application.

Model deployment is extremely important.

Once you’ve created a machine learning model capable of predicting some key aspects of

the future, you should make that model 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 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.

Another way to deploy SparkML models is by using MLeap.

Finally, 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 is another crucial step.

Once you’ve deployed a machine learning model, you need to keep track of its prediction

performance as new data arrives in order to maintain outdated models.

Following are some examples of model monitoring tools:

ModelDB is 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.

A generic, multi-purpose tool called Prometheus is also widely used for machine learning model

monitoring, although it’s not specifically made for this purpose.

Model performance is not exclusively measured through accuracy.

Model bias against protected groups like gender or race is also important.

The IBM AI Fairness 360 open source toolkit does exactly this.

It detects and mitigates against bias in machine learning models.

Machine learning models, especially neural-network-based deep learning models, can be subject to adversarial

attacks, where an attacker tries to fool the model with manipulated data or by manipulating

the model itself.

The IBM Adversarial Robustness 360 Toolbox can

be used to detect vulnerability to adversarial attacks and help make the model more robust.

Machine learning modes are often considered to be a black box that applies some mysterious

“magic.”

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.

The IBM AI Explainability 360 Toolkit can also illustrate training for a simpler machine

learning model by explaining how different input variables affect the final decision

of the model.

Options for code asset management tools have been greatly simplified:

For code asset management – also referred to as version management or version control

– Git is now the standard.

Multiple services have emerged to support Git, with the most prominent being GitHub,

which provides hosting for software development version management.

The runner-up is definitely GitLab, which has the advantage of being a fully open source

platform that you can host and manage yourself.

Another choice is Bitbucket.

Data asset management, also known as data governance or data lineage, is another crucial

part of enterprise grade data science.

Data has to be versioned and annotated with metadata.

Apache Atlas is a tool that supports this task.

Another interesting project, ODPi Egeria, is 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.

Finally, Kylo is an open source data lake management software platform that provides

extensive support for a wide range of data asset management tasks.

This concludes part one of this two-part series.

Now let’s move on to part two.

Welcome to part two of this series.

In this section, we’ll cover the development environment, open source data integration,

transformation, and visualization tools.

One of the most popular current development environments that data scientists are using

is “Jupyter.”

Jupyter first emerged as a tool for interactive Python programming; it now supports more than

a hundred different programming languages through “kernels.”

Kernels shouldn’t be confused with operating system 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.

From a user’s perspective, the main difference introduced by JupyterLab is the ability to

open different types of files, including Jupyter Notebooks, data, and terminals.

You can then arrange these files on the canvas.

Although Apache Zeppelin has been fully reimplemented, it’s 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 Apache Zeppelin,

and plotting doesn’t require coding.

You can also extend these capabilities by using additional libraries.

RStudio is one of the oldest development environments for statistics and data science, having been

introduced in 2011.

It exclusively runs R and all associated R libraries.

However, Python development is possible and R is therefore tightly integrated into this

tool to provide an optimal user experience.

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.

Although Spyder does not have the same level of functionality as RStudio, data scientists

do consider it an alternative.

But in the Python world, Jupyter is used more frequently.

This diagram shows how Spyder integrates code, documentation, visualizations, and other components

into a single canvas.

Sometimes your data doesn’t fit into a single computer’s storage or main memory capacity.

That’s where cluster execution environments come in.

The well known cluster-computing framework Apache Spark is among the most active Apache

projects and is used across all industries, including in many Fortune 500 companies.

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.

After Apache Spark began to gain market share, Apache Flink was created.

The key difference between Apache Spark and Apache Flink is that Apache Spark is a batch

data processing engine, capable of processing huge amounts of data file by file.

Apache Flink, on the other hand, is a stream processing image, with its main focus on processing

real-time data streams.

Although engine supports both data processing paradigms, Apache Spark is usually the choice

in most use cases.

One of the latest developments in the data science execution environments is called “Ray,”

which has a clear focus on large-scale deep learning model training.

Let’s look at open source tools for data scientists that are fully integrated and visual.

With these tools, no programming knowledge is necessary.

Most important tasks are supported by these tools; these tasks include data integration,

transformation, data visualization, and model building.

KNIME originated at the University of Konstanz in 2004.

As you can see, KNIME has a visual user interface with drag-and-drop capabilities.

It also has built-in visualization capabilities.

Knime can be be extended by programming in R and Python, and has connectors to Apache

Spark.

Another example of this group of tools is Orange.

It’s less flexible than KNIME, but easier to use.

In this video, you’ve learned about the most common data science tasks and which open

source tools are relevant to those tasks.

In the next video, we’ll describe some established commercial tools that you’ll encounter in

your data science experience.

Let’s move on to the next video to get more details.

We previously covered open source tools for data science.

Now, let’s look at the commercial options you’ll find in many enterprise projects.

Let’s revisit our overview of different tool categories.

In data management, most of an enterprise’s relevant data is stored in an

Oracle Database, Microsoft SQL Server, or IBM Db2.

Although open source databases are gaining popularity, those three data management products

are still considered the industry-standard.

They won’t disappear in the near future.

It’s not just about functionality.

Data is at the heart of every organization, and the availability of commercial supports

plays a major role.

Commercial supports are delivered directly from software vendors, influential partners,

and support networks.

When we focus on commercial data integration tools, we’re talking about “extract, transform,

and load,” or “ETL” tools.

According to a Gartner Magic Quadrant, Informatica Powercenter and IBM InfoSphere DataStage are

the leaders, followed by products from SAP, Oracle, SAS, Talend, and Microsoft.

These tools support design and deployment of ETL data-processing pipelines through a

graphical interface.

They also provide connectors to most of the commercial and open source target information

systems.

Finally, Watson Studio Desktop includes a component called Data Refinery, which enables

the defining and execution of data integration processes in a spreadsheet style.

In the commercial environment, data visualizations are utilizing business intelligence, or “BI”,

tools.

Their main focus is to create visually attractive and easy-to-understand reports and live dashboards.

The most prominent commercial examples are: Tableau, Microsoft Power BI, and IBM Cognos

Analytics.

Another type of visualization targets data scientists rather than regular users.

A sample problem might be “How can different columns in a table relate to each other?”

This type of functionality is contained in Watson Studio Desktop.

If you want to build a machine learning model using a commercial tool, you should consider

using a data mining product.

The most prominent of these types of products are: SPSS Modeler and SAS Enterprise Miner.

In addition, A version of SPSS Modeler is also available in Watson Studio Desktop, based

on the cloud version of the tool.

We’ll talk more about cloud-based tools in the next video.

In commercial software, model deployment is tightly integrated in the model building process.

This diagram shows an example of the SPSS Collaboration and Deployment Services which

are used to deploy any type of asset created by the SPSS software tools suite.

Other vendors use the same type of process.

Commercial software can also export models in an open format.

For example, SPSS Modeler supports the exporting of models as Predictive Model Markup Language,

or PMML, which can be read by many other commercial and open software packages.

Model monitoring is a new discipline and there are currently no relevant commercial tools

available.

As a result, open source is the first choice.

The same is true for code asset management.

Open source with Git and GitHub is the effective 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 using metadata.

Vendors, including Informatica Enterprise Data Governance and IBM, provide tools for

these specific tasks.

The IBM InfoSphere Information Governance Catalog covers functions like data dictionary,

which facilitates discovery of data assets.

Each data asset is assigned to a data steward -- the data owner.

The data owner is responsible for that data asset and can be contacted.

Data lineage is also covered; this enables a user to track back through the transformation

steps followed 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.

It’s usually consumed through the cloud, and we’ll cover more about it in a later

lesson.

There is also a desktop version available.

Watson Studio Desktop combines Jupyter Notebooks with graphical tools to maximize data scientists’

performance.

Watson Studio, together with Watson Open Scale, is a fully integrated tool covering the full

data science life cycle and all the tasks we’ve discussed previously.

We’ll talk more about both in the next lesson.

but just keep in mind that they can be deployed in a local data center on top of Kubernetes

or RedHat OpenShift.

Another example of a fully integrated commercial tool is H2O Driverless AI, which covers the

complete data science life cycle.

In this lesson, you’ve learned how most common data science tasks are supported by

commercial tools.

In the next video, we’ll discover data science tools that are available exclusively on the

cloud.

Since we’ve previously covered open source tools for data science, let’s look at the

commercial options you’ll find in many enterprise projects.

Take another 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 especially holds true 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’ve changed the title here and added the word “Platform.”

These clusters are composed of multiple server machines, transparently for the user, in the

background.

Watson Studio, together with Watson OpenScale, covers the complete development life cycle

for all data science, machine learning, and AI tasks.

Another example is 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.

And finally, another example is H2O Driverless AI, which we’ve already introduced in the

last video.

Although it is a product that you download and install, one-click deployment is available

for the common cloud service providers.

Since operations and maintenance are not done by the cloud provider, as is the case with

Watson Studio, Open Scale, and Azure Machine Learning, this delivery model should not be

confused with Platform or Software as a Service -- PaaS or SaaS.

In data management, with some exceptions, there are SaaS versions of existing open source

and commercial tools.

Remember, SaaS stands for “software as a service.”

It means that the cloud provider operates the tool for you in the cloud.

As an example, the cloud provider operates the product by backing up your data and configuration

and installing updates.

As mentioned, there is proprietary tooling, which is only available as a cloud product.

Sometimes it’s only available from a single cloud provider.

One example of such a service is Amazon Web Services DynamoDB, a NoSQL database that allows

storage and retrieval of data in a key-value or a document store format.

The most prominent document data structure is JSON (pronounced “jay-sun”).

Another flavour of such a service is 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.

And IBM offers Db2 as a service as well.

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.

When it comes to commercial 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.

Two widely used commercial data integration tools are Informatica Cloud Data Integration

and 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.

The market for cloud data visualization tools is huge, and every major cloud vendor has

one.

An example of a smaller company’s cloud-based data visualization tool is DataMeer.

IBM offers it’s famous Cognos Business intelligence suite as cloud solution as well.

IBM Data Refinery also offers data exploration and visualization functionality in Watson

Studio.

Again, these are just some examples of a rapidly changing and growing commercial ecosystem

among a huge number of established and emerging vendors.

In Watson Studio, an abundance of different visualizations can be used to better understand

data.

For example, this 3D bar chart enables you to visualize a target value on the vertical

dimension, which is dependent on two other values on the horizontal dimensions.

Coloring enables you to visualize a third dimension.

Hierarchical edge bundling enables you to visualize correlations and affiliations between

entities.

If sufficient, a classic bar chart can do the job as well, whereas a 2D scatter plot

with a heat map shows two dependent data fields, one on the y axis and one as color intensity.

A tree map shows distribution of subsets within a set, the famous pie chart does the same

but in a non-hierarchical manner, and finally, a word cloud pops out significant terms in

a document corpus.

Model building can be done using a service such as 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.

Nearly every cloud provider has a solution for this task.

Model deployment in commercial software is usually tightly integrated to the model building

process.

Here is an example of the SPSS Collaboration and Deployment Services, which can be used

to deploy any type of 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.

As an example, SPSS Modeler supports exporting models as Predictive Model Markup Language,

or “PMML,” which can be read by numerous other commercial and open software packages.

Watson Machine Learning can also be used to deploy a model and make it available to consumers

using a REST interface.

Amazon SageMaker Model Monitor is an example of a cloud tool that continuously monitors

deployed machine learning and deep learning models.

Again, every major cloud provider has similar tooling.

This is also the case for Watson OpenScale.

OpenScale and Watson Studio…

…unify the landscape.

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

We’ll cover Open Scale will be covered in a later video.

You’ve learned how the most common tasks in data science are supported by commercial

cloud tools.

Integration provides us the ability to use the same tools for multiple tasks.

In the next videos, we’ll look at packages, APIs, datasets, and models for data science.