**Overview of Data Science Tools**

In this module, you will learn about the different types and categories of tools that data scientists use and popular examples of each. You will also become familiar with Open Source, Cloud-based, and Commercial options for data science tools.

**Learning Objectives**

* Describe the components of a Data Scientist's toolkit and list various tool categories.
* List examples of open-source tools used in data science.
* Describe the relevant features of open-source tools.
* List examples of commercial and cloud-based data science tools.

# **Course Introduction**

## **Course Introduction**

The course introduction video provides an overview of what you will learn in the "Tools for Data Science" course. Here’s a simple breakdown:

* **Purpose**: The course is designed for anyone interested in becoming a data scientist, regardless of their programming background.
* **Content**: You will explore various tools used in data science, which help analyze data and draw conclusions.
* **Modules**: The course is divided into several modules, each focusing on different aspects of data science tools, such as:
  + Overview of tools
  + Programming languages used in data science (like Python and R)
  + Libraries and APIs
  + Working with Jupyter Notebooks and RStudio
  + A final project where you create and share your own Jupyter Notebook.

The goal is to give you hands-on experience with these tools so you can develop the skills needed for data science.

summary of the course introduction:

* **Course Title**: Tools for Data Science
* **Target Audience**: Anyone interested in becoming a data scientist, with or without programming knowledge.
* **Focus**: Learning about various tools used in data science to analyze data and derive insights.
* **Modules Include**:
  + Overview of data science tools
  + Programming languages (Python, R, etc.)
  + Libraries, APIs, and datasets
  + Jupyter Notebooks and RStudio
  + Final project to create and share a Jupyter Notebook

The course emphasizes hands-on experience to build practical skills in data science.

## **Learning Goals for the Course**

In this course you will be introduced to a Data Scientist's workbench or toolkit that consists of a variety of tools, languages, libraries, APIs, data sets, models, etc. used by Data Scientists. Try not to be overwhelmed by the sheer number of components and tools that exist in the Data Science ecosystem. The main goal of the course is for you to be knowledgeable about the kinds of tools Data Scientists use, their examples, and get some hands-on time with a few key tools.

As such, you are not required to recall the name of every single tool covered in the course. However, be familiar with the categories or types of tools and 1 or 2 examples each type. Modules 4 and 5 of this course will cover some of the most important ones for a beginner Data Scientist in greater depth and enable you to get hands-on experience with them. As you take additional Data Science courses, you will become more acquainted with some of the other tools and libraries. Some may be required to perform more specialized or advanced Data Science or Machine Learning tasks. So don't try to remember all of the names just now. Pay special attention, though, to Video and Lesson summaries.

To successfully complete the course, you are required to complete the first 6 out of the 7 modules in the course. The 7th module is an optional one.

Here is what you will be learning in each module:

**Module 1: Overview of Data Science Tools**

In this module, you will learn about the different types and categories of tools that data scientists use and popular examples of each. You will also become familiar with Open Source, Cloud-based, and Commercial options for data science tools.

* Describe the components of a Data Scientist's toolkit and list various tool categories
* List examples of Open Source, Commercial, and Cloud-based tools in various categories

**Module 2: Languages of Data Science**

This module will bring awareness about the criteria determining which language you should learn. You will learn the benefits of Python, R, SQL, and other common languages such as Java, Scala, C++, JavaScript, and Julia. You will explore how you can use these languages in Data Science. You will also look at some sites for more information about the languages.

**Learning Objectives**

* Identify the criteria and roles for determining the language to learn.
* Identify the users and benefits of Python.
* Identify the users and uses of the R language.
* Define SQL elements and list their benefits.
* Review languages such as Java, Scala, C++, JavaScript, and Julia.
* List the global communities for connecting with other users.

**Module 3: Packages, APIs, Data Sets and Models**

This module will give you in-depth knowledge of different libraries, APIs, dataset sources and models used by data scientist.

**Learning Objectives**

* List examples of the various libraries: scientific, visualization, machine learning, and deep learning.
* Define REST API to request and respond.
* Describe data sets and sources of data.
* Explore open data sets on the Data Asset eXchange.
* Describe how to use a learning model to solve a problem.
* List the tasks that a data scientist needs to perform to build a model.
* Explore ML models in the Model Learning eXchange.

**Module 4: Jupyter Notebooks and JupyterLab**

This module introduces the Jupyter Notebook and JupyterLab. You will learn how to work with different kernels and the basic Jupyter architecture. In addition, you will identify the tools in an Anaconda Jupyter environment. Finally, the module overviews cloud-based Jupyter environments and their data science features.

**Learning Objectives**

* Describe how to use the notebooks in JupyterLab.
* Describe how to work in a notebook session.
* Describe the basic Jupyter architecture.
* Describe how to work with kernels.
* Identify tools in Anaconda Jupyter environments.
* Describe cloud-based Jupyter environments and their data science features.

**Module 5: RStudio and GitHub**

This module will start with an introduction to R and RStudio and will end up with Github usage. You will learn about the different R visualization packages and how to create visual charts using the plot function.

Further in the module, you will develop the essential conceptual and hands-on skills to work with Git and GitHub. You will start with an overview of Git and GitHub, creating a GitHub account and a project repository, adding files, and committing your changes using the web interface. Next, you will become familiar with Git workflows involving branches, pull requests (PRs), and merges. You will also complete a project at the end to apply and demonstrate your newly acquired skills.

**Learning Objectives**

* Describe R capabilities and RStudio environment.
* Use the inbuilt R plot function.
* Explain version control and describe the Git and GitHub environment.
* Describe the purpose of source repositories and explain how GitHub satisfies the needs of a source repository.
* Create a GitHub account and a project repository.
* Demonstrate how to edit and upload files in GitHub.
* Explain the purpose of branches and how to merge changes.

**Module 6: Final Project and Assessment**

In this module, you will work on a final project to demonstrate some of the skills learned in the course. You will also be tested on your knowledge of various components and tools in a Data Scientist's toolkit learned in the previous modules.

**Learning Objectives**

* Create a Jupyter Notebook with markdown and code cells
* List examples of languages, libraries and tools used in Data Science
* Share your Jupyter Notebook publicly on GitHub
* Evaluate notebooks submitted by your peers using the provided rubric
* Demonstrate proficiency in Data Science toolkit knowledge

**Module7: IBM Watson Studio**

This is as an optional module if you are interested in learning about and working with data science tools from IBM such as Watson Studio.

**Learning Objectives**

* Find common resources in Watson Studio and IBM Cloud Pak for Data.
* Create an IBM Cloud account, service, and project in Watson Studio.
* Create and share a Jupyter Notebook.
* Use different types of Jupyter Notebook templates and kernels on IBM Watson Studio.
* Describe how to connect a Watson Studio account and publish a notebook in GitHub.

# **Data Science Tools**

**Data Management** in simple terms.

Data Management: A Simple Explanation  
Data management is like organizing a big library. Just as a librarian collects, stores, and retrieves books efficiently, data management involves collecting, storing, and retrieving data securely and effectively. This data can come from various sources, like social media, sensors, or online stores. The goal is to ensure that the data is available whenever you need it, just like how you can find a book in a well-organized library.

Imagine you have a treasure chest filled with different types of toys. If you just throw them in without any organization, it will be hard to find your favorite toy when you want to play with it. But if you sort them by type—like cars in one section and dolls in another—you can easily find what you’re looking for. Similarly, data management helps keep data organized so that it can be accessed quickly and efficiently.

summarize the key points from the video on **Categories of Data Science Tools** in simple terms:

1. **Data Science Tasks**: Data scientists perform various tasks, including:
   * **Data Management**: Collecting and storing data securely.
   * **Data Integration and Transformation (ETL)**: Extracting data from different sources, transforming it into a usable format, and loading it into a central storage (like a Data Warehouse).
   * **Data Visualization**: Creating visual representations of data (like charts and graphs) to help understand it better.
   * **Model Building**: Using algorithms to train a model that can make predictions based on data.
   * **Model Deployment**: Making the trained model available for use in real-world applications.
   * **Model Monitoring and Assessment**: Continuously checking the model's performance to ensure it remains accurate.
2. **Supporting Tools**:
   * **Data Asset Management**: Organizing and managing data collected from various sources.
   * **Code Asset Management**: Keeping track of changes in code and collaborating with others using platforms like GitHub.
   * **Execution and Development Environments**: Providing the necessary tools and resources to develop and run code.

The video emphasizes how these categories and tools work together to help data scientists effectively use raw data.

## **Open Source Tools for Data Science**

Data Integration and Transformation: A Simple Explanation

In the world of data science, data integration and transformation are like preparing ingredients before cooking a delicious meal. Imagine you have various ingredients (data) from different sources, and you need to combine and prepare them to create a tasty dish (insightful analysis). This process is often referred to as ETL, which stands for Extract, Transform, and Load. First, you **extract** the data from various sources, then you **transform** it into a format that makes sense, and finally, you **load** it into a system where it can be analyzed.

To make it even clearer, think of a fruit salad. You gather different fruits (extract), chop them into bite-sized pieces (transform), and then put them in a bowl (load) for everyone to enjoy. In data science, tools like Apache AirFlow and KubeFlow help automate this process, making it easier for data scientists to handle large amounts of data efficiently.

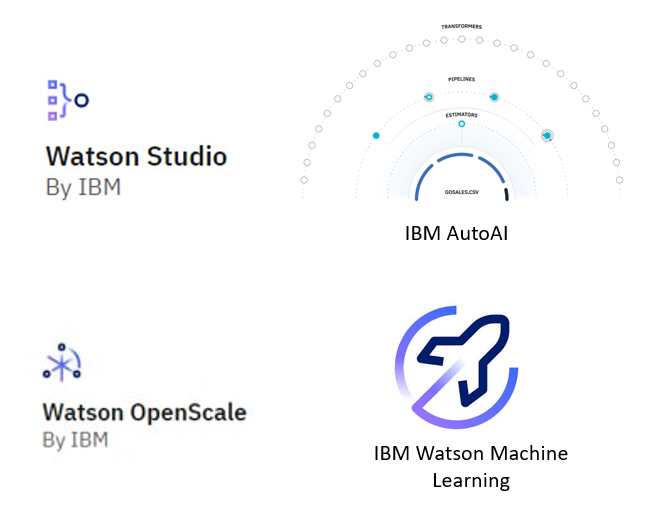
"Open-Source Tools for Data Science Part 1," you will learn about various open-source tools used in data science, including:

* **Data Management Tools**: MySQL, PostgreSQL, MongoDB, Apache CouchDB, Apache Cassandra, Hadoop File System, Ceph, and Elasticsearch.
* **Data Integration and Transformation Tools**: Apache AirFlow, KubeFlow, Apache Kafka, Apache Nifi, Apache SparkSQL, and NodeRED.
* **Data Visualization Tools**: Pixie Dust, Hue, Kibana, and Apache Superset.
* **Model Deployment Tools**: Apache PredictionIO, Seldon, MLeap, and TensorFlow services.
* **Model Monitoring Tools**: ModelDB, Prometheus, IBM AI Fairness 360, and IBM Adversarial Robustness 360 Toolbox.
* **Code Asset Management Tools**: Git, GitHub, GitLab, and Bitbucket.
* **Data Asset Management Tools**: Apache Atlas, ODPi Egeria, and Kylo.

The course emphasizes hands-on experience and will help you develop skills for working with these tools. If you have specific questions or need clarification on any topic

## **Model Deployment**

IBM offers various tools and platforms tailored for model development across various domains. Here are a few examples:



* **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.

## **Open Source Tools for Data Science**

concept of **Jupyter Notebooks** and **Jupyter Lab** in simple terms.

Jupyter Notebooks are like interactive notebooks for coding, where you can write code, see the results, and add notes all in one place. Imagine it as a digital notebook where you can mix your thoughts (notes), your calculations (code), and the results (outputs) all together. This makes it super easy to understand what you did and why you did it, just like how you might jot down your thoughts while solving a math problem in a physical notebook.

Now, Jupyter Lab is the next version of Jupyter Notebooks. Think of it as an upgraded version of your favorite notebook that allows you to open multiple pages at once, like having several notebooks open on your desk. You can arrange them however you like, making it easier to work on different tasks simultaneously. It’s more modern and flexible, allowing you to work with various types of files, not just code.

1. **Jupyter Notebooks**

* **Purpose**: Jupyter Notebooks are widely used for interactive programming, especially in Python. They allow data scientists to write code, run it, and see the results immediately.
* **Features**:
  + **Unified Document**: Combines code, output, visualizations, and documentation (Markdown) in a single document.
  + **Kernels**: Supports multiple programming languages through "kernels," allowing users to work in languages like R, Julia, and more.

2. **Jupyter Lab**

* **Overview**: Jupyter Lab is the next generation of Jupyter Notebooks, designed to be more flexible and powerful.
* **Key Differences**:
  + **Multiple File Types**: Users can open and arrange different types of files (e.g., notebooks, data files, terminals) on a single canvas.
  + **Modern Interface**: Offers a more modular and user-friendly interface compared to Jupyter Notebooks.

3. **RStudio**

* **Background**: RStudio is a well-established development environment specifically for R programming.
* **Features**:
  + **Integrated Environment**: Combines programming, execution, debugging, and data visualization in one tool.
  + **Python Support**: While primarily for R, it also supports Python development.

4. **Spyder**

* **Purpose**: Spyder is another development environment that mimics RStudio but is tailored for Python.
* **Functionality**: Integrates code, documentation, and visualizations into a single interface, although it may not have all the features of RStudio.

5. **Apache Spark**

* **Overview**: A powerful open-source engine for big data processing.
* **Key Property**: Known for its **linear scalability**, meaning that doubling the number of servers in a cluster roughly doubles performance.
* **Use Cases**: Widely used across industries, including Fortune 500 companies, for processing large datasets.

6. **Apache Flink**

* **Comparison with Spark**: Developed after Spark, Flink focuses on real-time data stream processing, while Spark is primarily a batch processing engine.
* **Capabilities**: Both can handle batch and stream processing, but Flink is optimized for real-time data.

7. **Ray**

* **Focus**: A newer tool designed for large-scale deep learning model training.
* **Use Case**: Particularly useful for training complex models that require significant computational resources.

8. **Visual Tools: KNIME and Orange**

* **KNIME**:
  + **Origin**: Developed at the University of Konstanz in 2004.
  + **User Interface**: Features a visual interface with drag-and-drop capabilities, making it user-friendly.
  + **Extensibility**: Can be extended with R and Python programming and has connectors to Apache Spark.
* **Orange**:
  + **Overview**: Another visual programming tool for data science.
  + **Ease of Use**: Less flexible than KNIME but simpler for beginners to use.

Conclusion

The video provides an overview of various open-source tools that data scientists can use, highlighting their features, strengths, and specific use cases. Understanding these tools is essential for anyone looking to work in data science, as they form the foundation of the data scientist's toolkit.

## **Commercial Tools for Data Science**

simplified overview of the key points from the video on **commercial tools for data science**:

1. **Data Management Tools**:
   * These tools help store and manage data. Common examples include **Oracle Database**, **Microsoft SQL Server**, and **IBM Db2**. They are considered industry standards.
2. **Data Integration Tools**:
   * These tools are used for **Extract, Transform, Load (ETL)** processes. They help gather data from various sources, process it, and load it into a central system. Notable tools include **Informatica PowerCenter** and **IBM InfoSphere DataStage**.
3. **Data Visualization Tools**:
   * These tools create visual reports and dashboards to help understand data better. Popular tools are **Tableau**, **Microsoft Power BI**, and **IBM Cognos Analytics**.
4. **Model Building Tools**:
   * For building machine learning models, tools like **SPSS Modeler** and **SAS Enterprise Miner** are commonly used. They help in developing predictive models.
5. **Model Deployment and Monitoring**:
   * Tools like **SPSS Collaboration and Deployment Services** help deploy models. However, model monitoring is still developing, with open-source tools being preferred.
6. **Data Asset Management**:
   * This involves managing data governance and lineage. Tools from **Informatica** and **IBM** help track data assets and ensure compliance with regulations.
7. **Integrated Development Environments**:
   * **Watson Studio** is an example of a fully integrated tool that covers the entire data science lifecycle, combining various functionalities for data scientists.

In summary, the video highlights various commercial tools that support different tasks in data science, from data management to visualization and model deployment. Each tool plays a crucial role in helping data scientists work efficiently and effectively.

## **Cloud Based Tools for Data Science**

**cloud-based tools for data science**.

Cloud-based tools for data science are like having a powerful toolbox that you can access from anywhere, as long as you have an internet connection. Imagine you have a magic toolbox that not only contains all the tools you need for your projects but also allows you to work with others seamlessly. These tools help you collect, analyze, and visualize data without needing to worry about the technical details of how everything works behind the scenes. For example, platforms like Watson Studio and Microsoft Azure Machine Learning provide everything you need to build and deploy machine learning models, just like a chef has all their cooking utensils in one place.

To make it even more relatable, think of these cloud tools as a community garden. In this garden, everyone can plant their seeds (data), water them (analyze), and eventually harvest the fruits (insights) together. The cloud provider takes care of the soil, watering, and maintenance, so you can focus on growing your plants without worrying about the hard work behind the scenes. This integration of tools allows data scientists and engineers to work more efficiently and collaboratively.

summary of **cloud-based tools for data science**:

* **Definition**: Cloud-based tools are online platforms that provide resources for data science tasks, allowing users to access powerful tools from anywhere with an internet connection.
* **Integration**: These tools often combine multiple functions, enabling users to collect, analyze, and visualize data seamlessly.
* **Examples**: Notable platforms include Watson Studio, Microsoft Azure Machine Learning, and H2O Driverless AI, which support the entire development life cycle of data science projects.
* **Benefits**: Users can focus on their work without worrying about technical maintenance, as the cloud provider handles operations like updates and backups.

## **Summary: Open Source Tools for Data Science**

**Objective**

This reading provides a summary of key open-source tools for Data Science covered in the Part 1 and Part 2 videos of this course.

They are broadly classified as -

* **Data Management Tools** - Facilitates the storage, organization, and retrieval of data. Includes Relational Databases, NoSQL Databases, and Big Data platforms.
* **Data Integration and Transformation Tools** - Streamlines data pipelines and automate data processing workflows. Task of data integration and transformation in the classic data warehousing world is to Extract, Transform, and Load (ETL).
* **Data Visualization Tools**- Provides graphical representation of data and assist with communicating insights.
* **Model Deployment, Monitoring and Assessment Tools**- Supports the building, deploying, monitoring, and evaluation of data and machine learning models.
* **Data Asset Management Tools**- Organizes and manages data, enforce access controls, and ensure asset backups.
* **Code Development and Execution Tools** - ProvideS environments for developing, testing, and deploying code, offering computational resources to execute it.
* **Code Asset Management Tools** - Enables the storage and management of code, track changes, and support collaborative development.

**Data Management Tools**

**MySQL**

* Popular open source **relational database management system** (RDBMS)
* Uses structured query language (SQL) to manage and store data.
* Common uses:
  + Web applications
  + Data warehousing
  + E-commerce

**PostgreSQL**

* Powerful and open source **relational database management system** (RDBMS)
* Emphasizes extensibility and SQL compliance.
* Offers advanced features such as:
  + Support for JSON
  + Full-text search
  + Spatial data

**Apache CouchDB**

* Document-oriented **NoSQL** database
* Uses JSON to store data
* Highly scalable
* Fault-tolerant
* Easy to use

**MongoDB**

* Document-oriented **NoSQL** database
* Stores data in a flexible JSON
* Provides:
  + Scalability
  + High availability
  + Data distribution
* Suitable for modern web applications that handle large volumes of unstructured data

**Apache Cassandra**

* Highly scalable, distributed Document-oriented **NoSQL** database
* Can handle large amounts of structured and unstructured data across many commodity servers.
* Offers:
  + High availability
  + Fault tolerance
  + Tunable consistency levels
* Suitable for mission-critical applications

**Hadoop Distributed File System (HDFS)**

* Designed to work with large datasets like Apache Hadoop in a distributed computing environment
* High-throughput data processing by splitting files into blocks (default 128MB), and these blocks are distributed across multiple DataNodes
* Data is replicated across different DataNodes ensuring high availability and fault tolerance
* Scalable and efficient

**Ceph**

* Free, open source software-defined storage platform suitable for hybrid cloud environments
* Designed for modern data centers
* Provides highly scalable, unified storage system that can be used for object storage (like AWS S3), block storage (like virtual disks for VMs), and file storage (like NFS) under one unified system
* High performance, availability and reliability

**Elasticsearch**

* Primarily a distributed RESTful search engine and analytics tool
* Based on the Lucene library.
* Full-text search, real-time data analytics
* Highly scalable
* Easy to use
* Powerful querying capabilities
* Real-time data indexing for fast document retrieval.

**Data Integration and Transformation Tools**

**Apache Airflow**

* Open-source platform for programmatically authoring, scheduling, and monitoring workflows
* Created originally by Airbnb
* Allows users to define and execute complex workflows
* Support for:
  + Task dependencies
  + Parallelism
  + Error handling

**Kubeflow**

* An open-source machine learning toolkit that allows execution of data science pipelines on top of Kubernetes.
* Provides a platform for building, deploying, and managing end-to-end machine learning workflows at scale
* Support for:
  + Distributed training
  + Model serving
  + Hyperparameter tuning

**Apache Kafka**

* Distributed streaming platform that allows applications to publish, process, and subscribe to streams of records in real-time
* Created originally from LinkedIn.
* It is scalable, fault-tolerant, and high-throughput
* Suitable for building mission-critical, data-intensive applications

**Apache NiFi**

* An open-source data integration platform that allows users to automate the flow of data between systems
* Provides a web-based user interface for designing and managing data flows
* Support for:
  + Data routing
  + Transformation
  + Enrichment
  + Among other capabilities

**Apache Spark SQL**

* A module in the Spark ecosystem that provides a programming interface for working with structured data using:
  + SQL
  + Data frames
  + Datasets
* Supports a wide range of data sources and provides optimized performance for complex data processing tasks.

**Node-RED**

* An open-source visual programming tool for wiring together hardware devices, APIs, and online services
* Allows users to create event-driven flows of messages
* low in resource consumption that it even runs on tiny devices like a Raspberry Pi.
* Support for:
  + Data transformation
  + Filtering
  + Aggregation

**Data Visualization Tools**

**PixieDust**

* Open-source library for creating interactive, exploratory data visualizations in Python and Jupyter notebooks
* Provides a range of built-in visualizations and data connectors
* Support for customization and extensibility through third-party libraries

**Hue**

* Open-source web interface for analyzing and visualizing large datasets in Apache Hadoop
* Offers a user-friendly experience for exploring data and creating visualizations
* No need for programming skills; can create visualizations from SQL queries

**Kibana**

* Open-source data visualization tool that allows users to interact with their data through a web-based interface
* Commonly used with Elasticsearch to analyze and visualize large datasets

**Apache Superset**

* A modern, enterprise-ready business intelligence web application that makes it easy to visualize and explore large datasets
* Offers a rich set of data visualization options, including:
  + Charts
  + Tables
  + Maps
  + Geospatial analysis
  + Real-time data processing

**Model Deployment Tools**

**Apache PredictionIO**

* Open-source machine learning server built on a scalable and distributed infrastructure
* Allows developers to quickly build, evaluate, and deploy predictive engines for various use cases such as:
  + Recommendation
  + Classification
  + Clustering

**Kubernetes**

* Open-source platform for container orchestration
* Automatically launches, scales, and manages containerized applications
* Offering features like:
  + Automatic scaling
  + Self-healing
  + Load balancing
* Enables the management and orchestration of containers across numerous hosts

**Apache Seldon**

* Open-source platform for deploying and managing machine learning models on Kubernetes
* Provides a way to:
  + Serve models at scale
  + Automate model deployment workflows
  + Monitor the performance of deployed models in real-time

**MLeap**

* Open-source library for serializing and deserializing learning models in a cross-platform file
* Gives users the ability to export models from different machine learning libraries and frameworks, such as:
  + Spark
  + Scikit-learn
  + TensorFlow
* Implements them in high-throughput, low-latency production environments

**TensorFlow Lite**

* Open-source tool for running machine learning models on mobile and embedded devices
* Allows effective inference on mobile and embedded platforms
* Supports a variety of hardware accelerators such as:
  + CPUs
  + GPUs
  + Custom ASICs

**Red Hat OpenShift**

* Container application framework based on Kubernetes
* With characteristics like automation, scalability, and security
* Offers a method for creating, deploying, and managing containerized applications

**TensorFlow Serving**

* Open-source utility that serves machine learning models in real-world settings
* Supports both HTTP and gRPC interfaces for serving predictions
* Provides high scalability and low latency deployment and management of TensorFlow models

**TensorFlow.js**

* Open-source library for building and deploying machine learning models in JavaScript
* Allows you to train and execute models directly in the browser or on Node.js
* Supports a wide range of model architectures, including neural networks, decision trees, and k-nearest neighbors

**Model Monitoring and Assessment Tools**

**ModelDB**

* Open-source platform for managing machine learning models and experiments
* Provides a way to track and reproduce experiments, version models, and collaborate with team members

**Prometheus**

* Freely available monitoring system that collects and stores metrics in real-time from different sources
* Allows you to visualize and set alerts on the health and performance of systems and apps
* Supports a variety of data gathering methods, such as HTTP endpoints, exporters, and agents

**IBM AI Fairness 360**

* Open-source toolkit for detecting and mitigating bias in machine learning models
* Provides a way to measure the fairness and bias of models, as well as a set of algorithms for mitigating bias and creating fairer models

**IBM AI Explainability 360**

* Open-source toolkit for explaining the behavior and decisions of machine learning models
* Provides a way to measure the explainability and interpretability of models, as well as a set of algorithms for generating explanations and visualizations of model behavior

**IBM Adversarial Robustness 360 Toolbox**

* Free and open-source library for protecting machine learning models from adversarial attacks
* Includes a method for measuring model robustness and vulnerability
* Includes a set of algorithms for improving model robustness and detecting adversarial examples

**Code Development and Execution Tools**

**Jupyter IDE**

* Open-source effort
* Supports:
  + Julia
  + Python
  + R development with Jupyter Notebook
  + JupyterLab
  + JupyterHub
* Create and share documents containing:
  + Live code
  + Equations
  + Visualizations
  + Narrative text
* JupyterLab includes:
  + Customized notebook organization
* JupyterHub extends all these capabilities to the enterprise

**RStudio**

* For developers
* Free and open-source IDE
* Built to manage and execute R code
* Works on all platforms
* Includes:
  + Version control
  + Project management capabilities

**Microsoft Visual Studio**

* An IDE that supports a variety of programming languages, including:
  + C
  + C++
  + C++/CLI
  + Visual Basic.NET
  + C#
  + F#
  + JavaScript
  + TypeScript
  + XML
  + XSLT
  + HTML
  + CSS
* Using plug-ins, supports:
  + Python
  + Ruby
  + Node.js
  + M
  + Other languages

**PyCharm**

* Primarily a subscription-based IDE environment
* Offers 16+ additional tools for coding assistance, testing, and web development
* Supports scientific development with IPython integration and Matplotlib and NumPy support
* Also offers a free community-based, open-source IDE with limited capabilities

**Spyder**

* Free, open-source Python-based IDE designed by and for scientists, engineers, and data analysts
* Features a unique combination of comprehensive development tools for:
  + Advanced editing
  + Analysis
  + Debugging
  + Profiling
  + Visualization capabilities

**Anaconda Navigator**

* Open-source GUI-based Navigator that supports Python development and integrates with:
  + Eclipse and PyDev
  + IDLE
  + IntelliJ
  + Microsoft Visual Studio Code (VS Code)
  + Ninja IDE
  + PyCharm
  + Python for Visual Studio Code
  + Python Tools for Visual Studio (PTVS)
  + Spyder
  + Sublime Text
  + Wing IDE

**Code Asset Management Tools**

**Git**

* Open-source version control system for tracking changes in code and collaboration among developers
* Provides a way to manage and organize code changes, collaborate on code development, and maintain a history of code revisions

**GitLab**

* Web-based Git repository manager
* Provides a complete DevOps platform for:
  + Source code management
  + Continuous integration and deployment
  + Monitoring
* Enables teams to collaborate on:
  + Code development
  + Automate build and deployment processes
  + Track metrics and performance across the entire software development lifecycle

**GitHub**

* Web-based Git repository hosting service that provides a platform for developers to collaborate on code and manage software projects
* Enables users to:
  + Create, fork, and contribute to open source projects
  + Track changes in code
  + Manage issues
  + Pull requests

**Bitbucket from Atlassian**

* Web-based Git repository hosting service
* Provides a platform for developers to collaborate on code and manage software projects, with features like:
  + Pull requests
  + Code review
  + Branch permissions

# **Module 1 Summary**

**Module 1 Summary**

Congratulations! You have completed this module. At this point in the course, you know:

* The Data Science Task Categories include:
  + Data Management - storage, management and retrieval of data
  + Data Integration and Transformation - streamline data pipelines and automate data processing tasks
  + Data Visualization - provide graphical representation of data and assist with communicating insights
  + Modelling - enable Building, Deployment, Monitoring and Assessment of Data and Machine Learning models
* Data Science Tasks support the following:
  + Code Asset Management - store & manage code, track changes and allow collaborative development
  + Data Asset Management - organize and manage data, provide access control, and backup assets
  + Development Environments - develop, test and deploy code
  + Execution Environments - provide computational resources and run the code

The data science ecosystem consists of many open source and commercial options, and include both traditional desktop applications and server-based tools, as well as cloud-based services that can be accessed using web-browsers and mobile interfaces.

**Data Management Tools**: include Relational Databases, NoSQL Databases, and Big Data platforms:

* MySQL, and PostgreSQL are examples of Open Source Relational Database Management Systems (RDBMS), and IBM Db2 and SQL Server are examples of commercial RDBMSes and are also available as Cloud services.
* MongoDB and Apache Cassandra are examples of NoSQL databases.
* Apache Hadoop and Apache Spark are used for Big Data analytics.

**Data Integration and Transformation Tools:** include Apache Airflow and Apache Kafka.

**Data Visualization Tools:** include commercial offerings such as Cognos Analytics, Tableau and PowerBI and can be used for building dynamic and interactive dashboards.

**Code Asset Management Tools:** Git is an essential code asset management tool. GitHub is a popular web-based platform for storing and managing source code. Its features make it an ideal tool for collaborative software development, including version control, issue tracking, and project management.

**Development Environments:** Popular development environments for Data Science include Jupyter Notebooks and RStudio.

* Jupyter Notebooks provides an interactive environment for creating and sharing code, descriptive text, data visualizations, and other computational artifacts in a web-browser based interface.
* RStudio is an integrated development environment (IDE) designed specifically for working with the R programming language, which is a popular tool for statistical computing and data analysis.