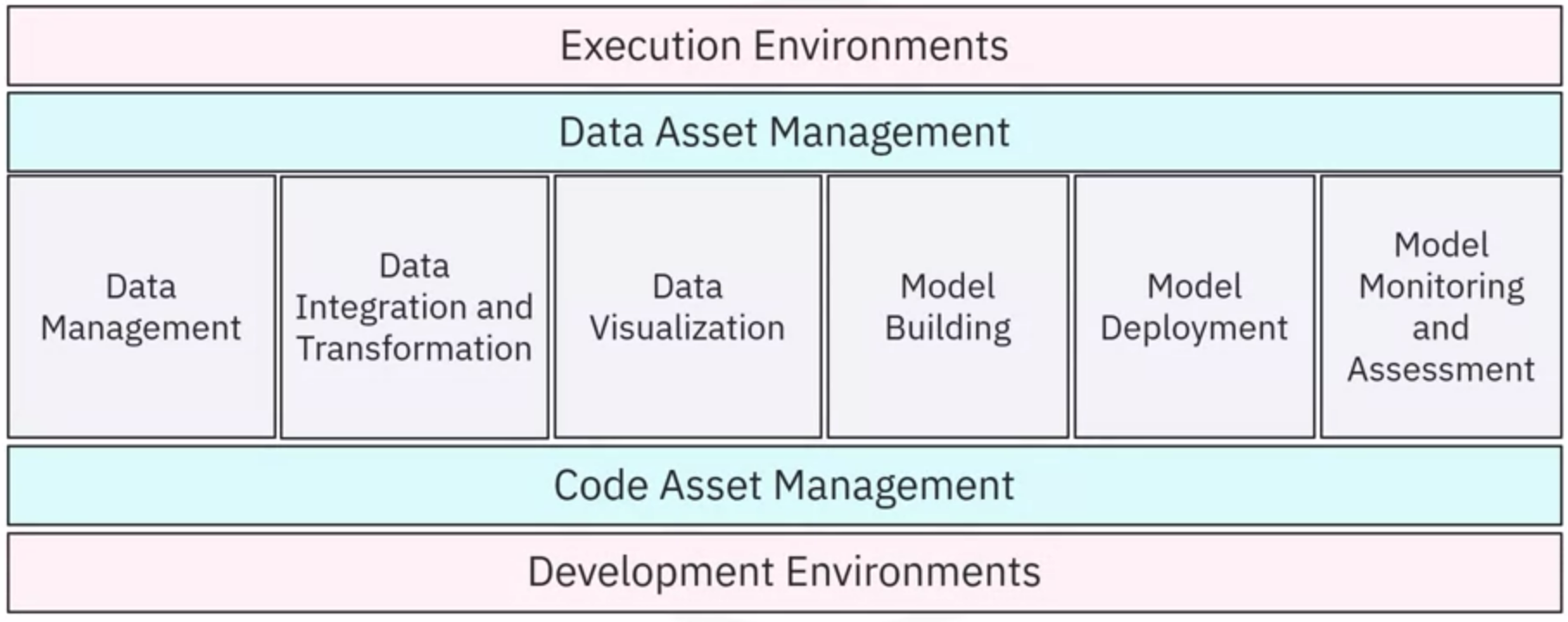
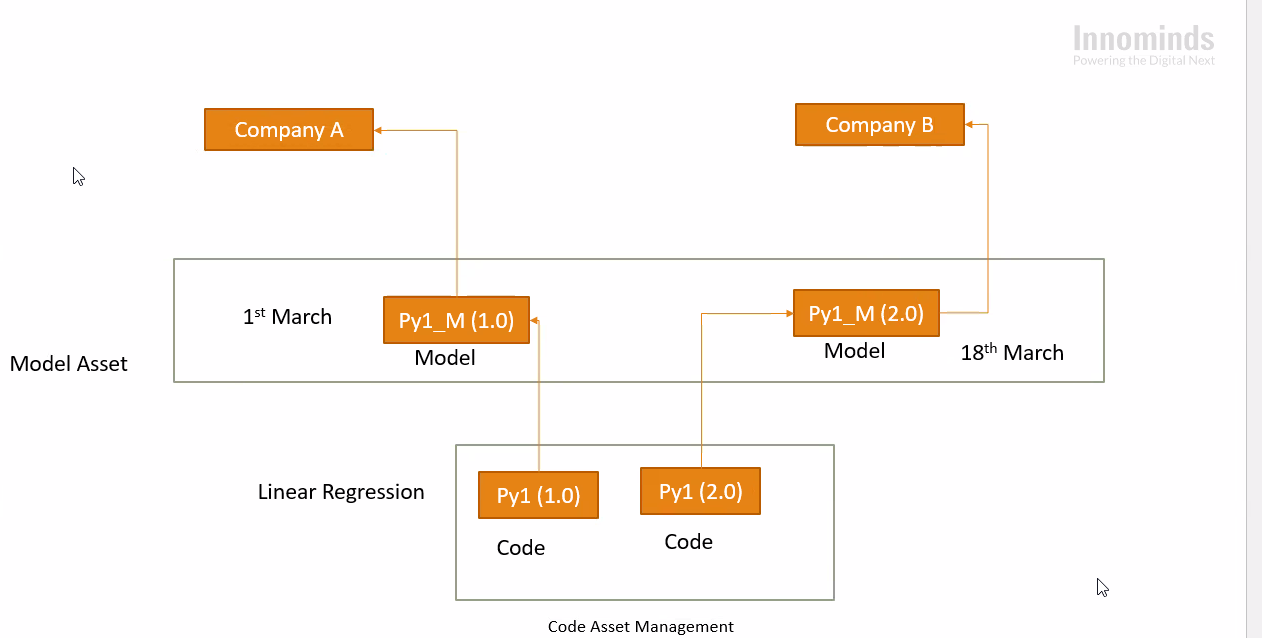
**Week 1:**

**Categories of Data Science tools:**



What is Code Asset management?

Example:

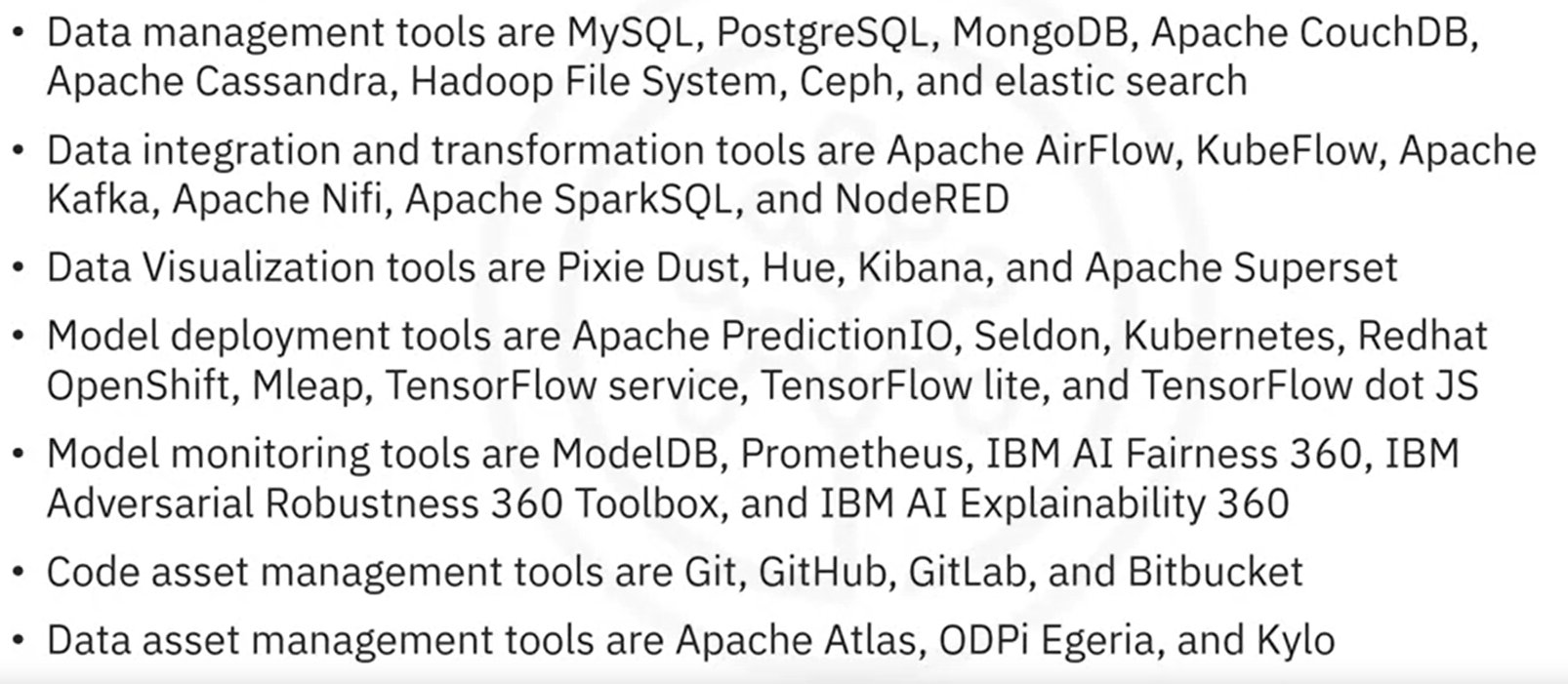


For maintaining versions of the same code for different companies (having different requirements with common similarities). For this purpose something called **code registry** is used in companies.

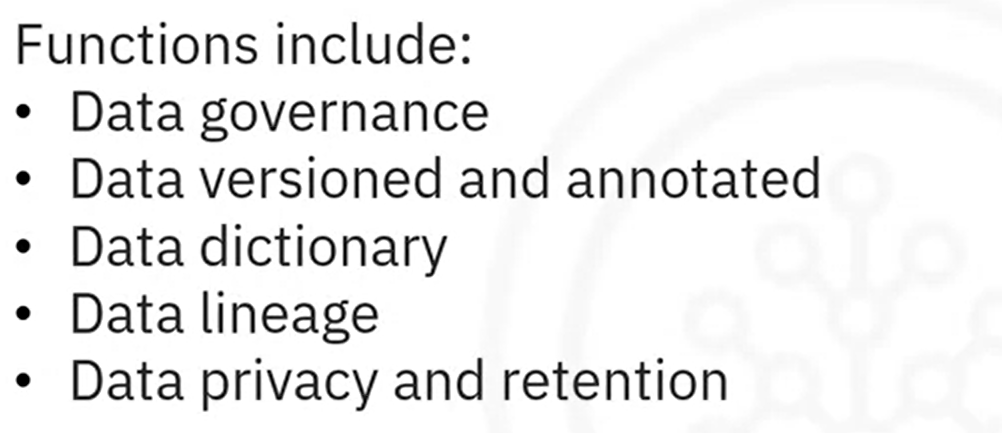
Data Science Tasks are supported by:

* Code asset management
* Data asset management
* Execution environment (like Watson Studio)
* Development environment

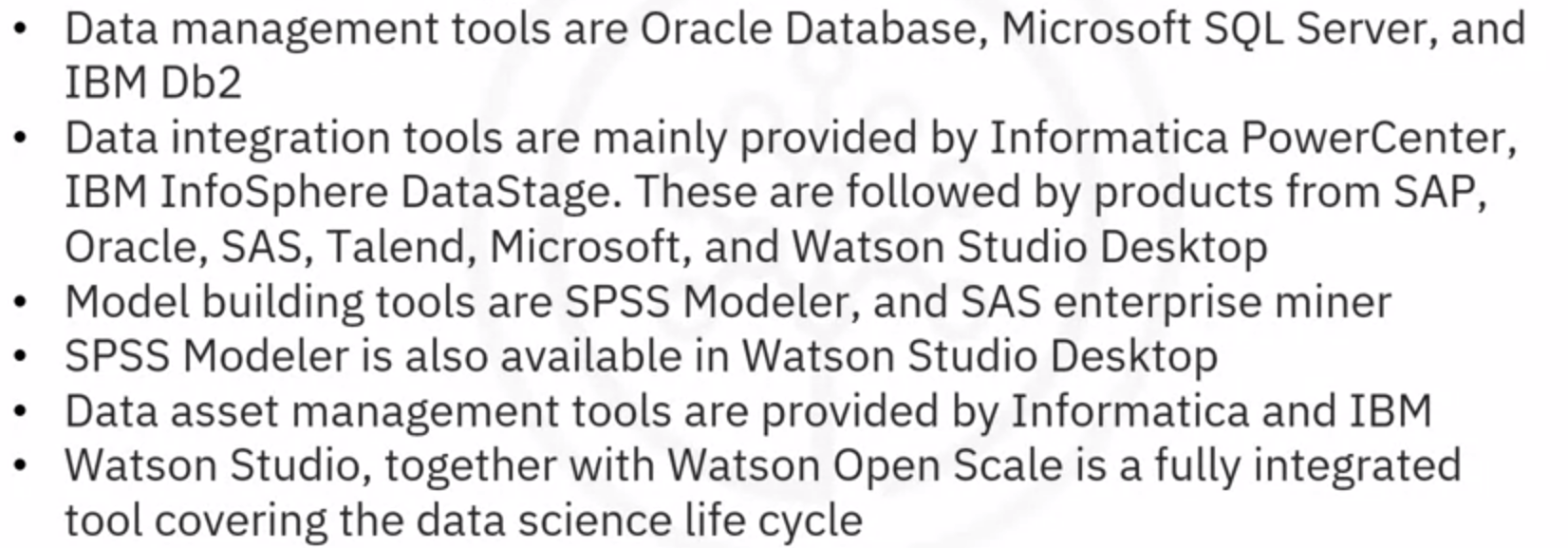
**Open source tools:**



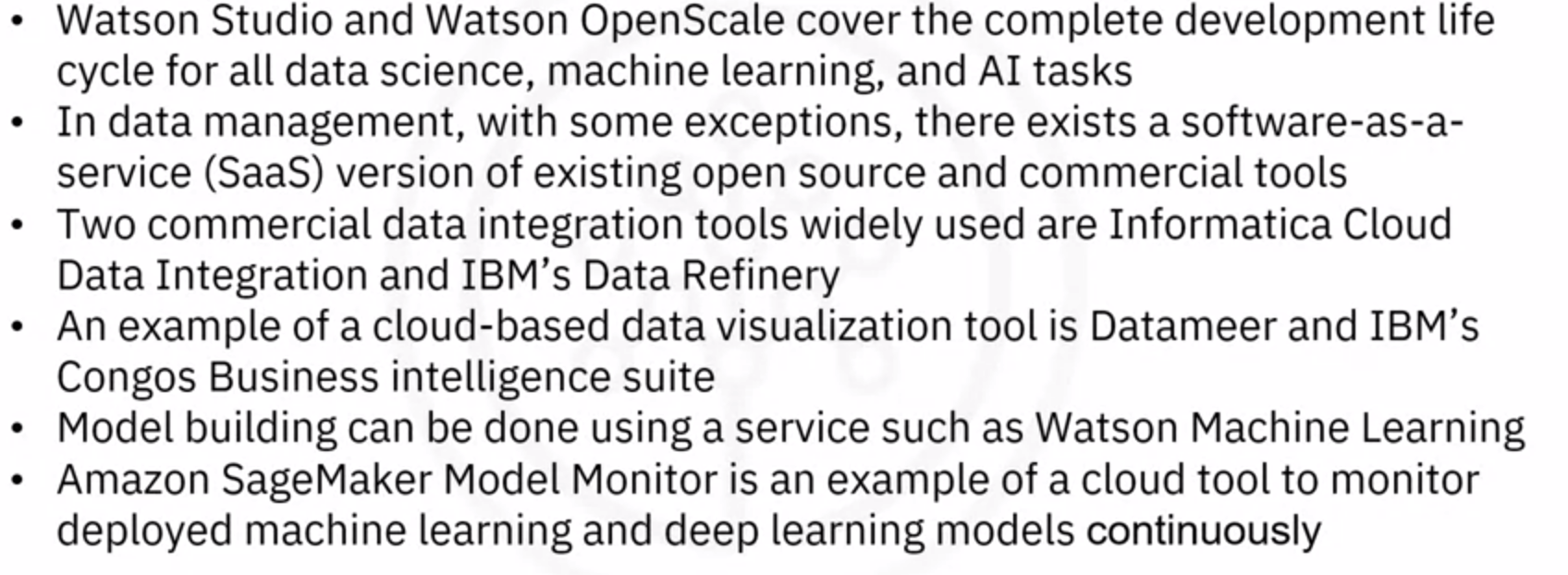
**Data asset management:**



**Commercial tools for data science:**



**Cloud based tools for data science:**

****

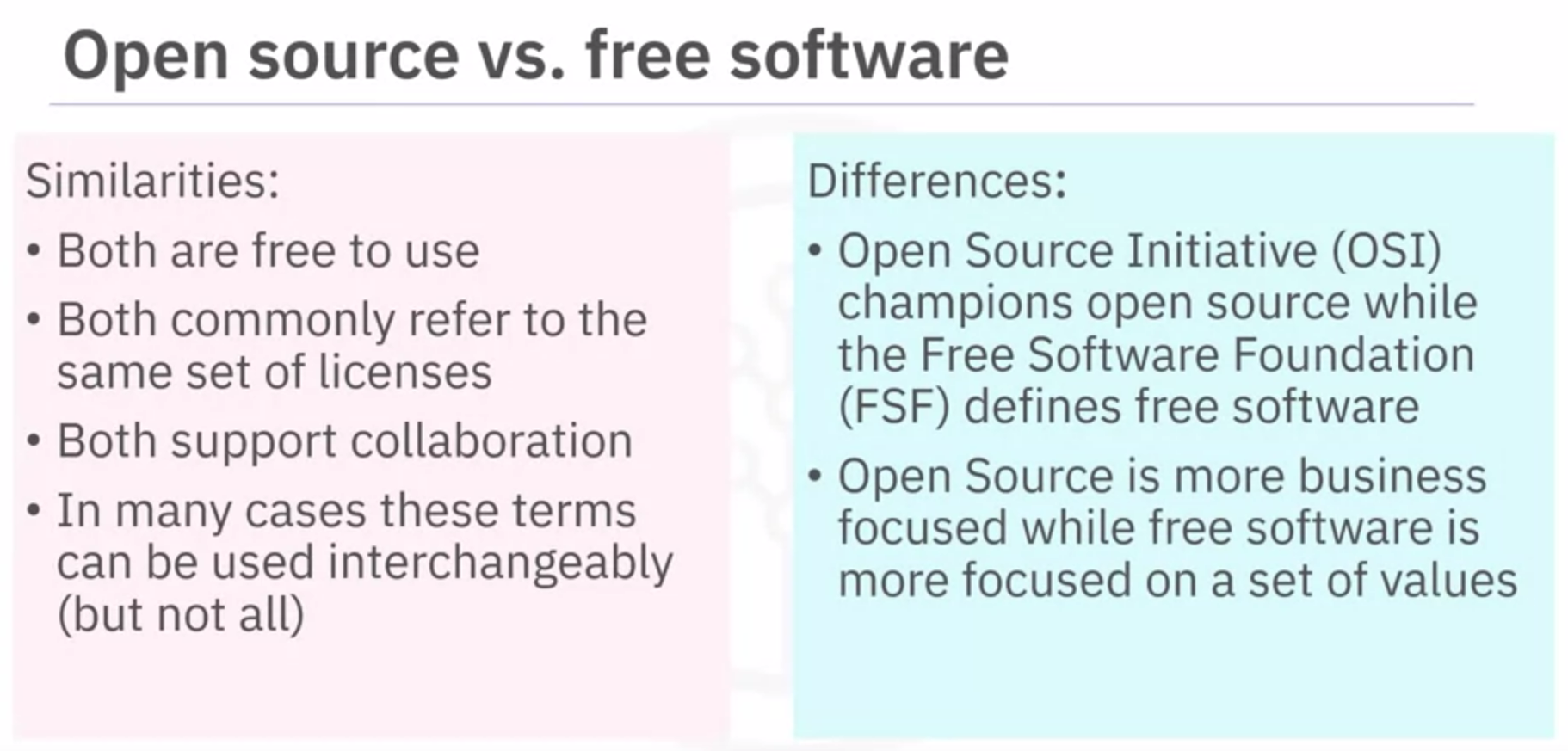
Watson Studio Desktop includes a component called Data Refinery that allows for the definition and execution of data integration processes in a spreadsheet-style.

Tools integrate code, documentation, and visualizations into a single canvas include Spyder and Jupyter notebook

**Week 2:**

**Languages of Data Science:**

* Language should be selected based on your needs, problems to be solved and company needs
* Languages used in data science include python, R, SQL, Scala, Java, C++, Julia, PHP, JS, GO, Ruby and Visual Basic



Python is open source while R is free software. For data science, Python's scientific computing libraries like Pandas, NumPy, SciPy, and Matplotlib are used. Python can also be used for Natural Language Processing (NLP) using the Natural Language Toolkit (NLTK)

SQL is non-procedural language. Scope is limited to querying and managing data in relational databases. It is not a data science language. It is an ANSI standard.

*Note: Data in tables are called structured data. Unstructured data include images and videos etc. SQL data is present in RDBMS while NOSQL is present in tables with no relationships (no primary and foreign keys) and we use MongoDB, Cassandra, CouchDB to handle such data.*

Data science Java tools include Weka (for data mining), Java-ML (a ml library), Apache MLlib (scalable ml) and Deeplearning4

Hadoop manages data processing and storage for big data applications running in clustered systems

Scala (scalable language) designed as extension to java. Apache Spark program is built using Scala for data science to be faster than Hadoop. Apache spark include Shark, MLlib, GraphX and Spark Streaming.

JavaScript applications for data science include TensorFlow.js which makes machine learning and deep learning possible in Node.js and in browser. JS is not related to Java. Tensorflow.js is adapted by open source libraries such as brain.js and machinelearn.js. R-js makes linear algebra possible in typescript (superset of javascript)

For data science, TensorFlow, MongoDB and Caffe were built using C++

Julia is a young language designed for high performance numerical analysis and computational science which executes directly on the processor. For data science, JuliaDB package is used for large persistent data sets.

**Week 3:**

**Packages, APIs, Datasets and Models:**

Python libraries in data science –

* Scientific computing libraries include pandas (data structures and tools), numpy (array and matrices)

Note: Pandas is built on top of numpy

* Data visualization libraries include Matplotlib (plots and graphs), Seaborn (heat maps, time series, violin plots)
* ML and Deep learning libraries include Scikit-learn (ML statistical modeling using regression, classification, clustering), Keras (for deep learning neural networks), TensorFlow (low level framework used for deep learning production and deployment), PyTorch (deep learning regression and classification)
* Note: Scikit-learn is built on matplotlib, sci-py and numpy

Apache Spark is used as a general purpose cluster-computing framework – data is processed in parallel multiple computers (cluster of computers) simultaneously. Apache spark functions similar to pandas, numpy and scikit-learn. Data can processed in python, R, Scala and SQL.

Scala libraries in data science and data engineering –

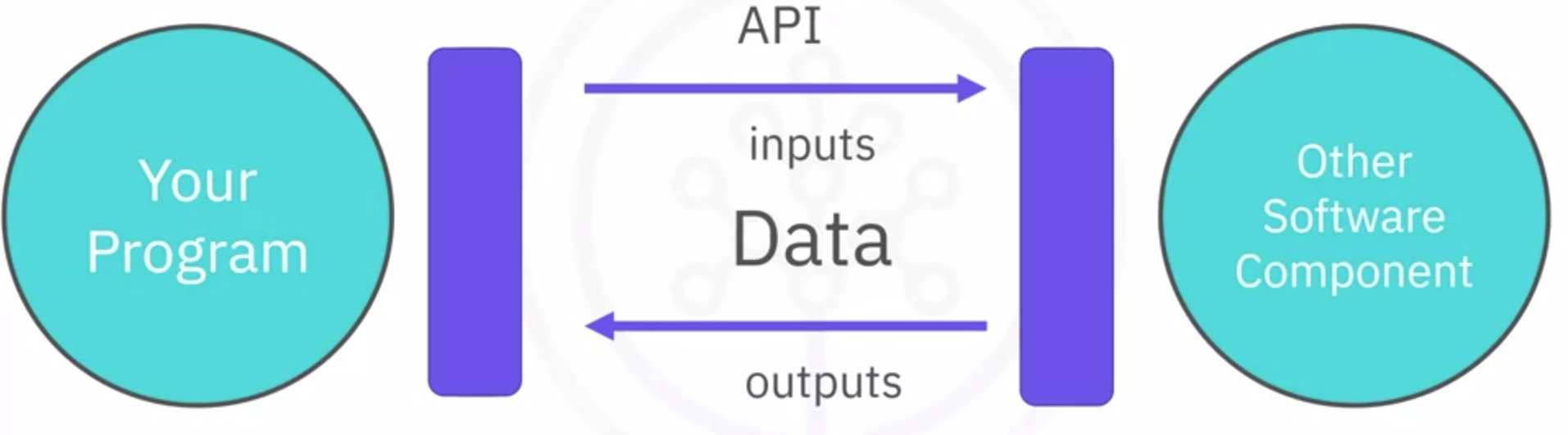
* Vegas (statistical data visualizations), Big DL (for deep learning)

R libraries for data science –

* Ggplot2 (for data visualization), libraries are allow to interface with Keras and TensorFlow

API (Application Programming Interface):

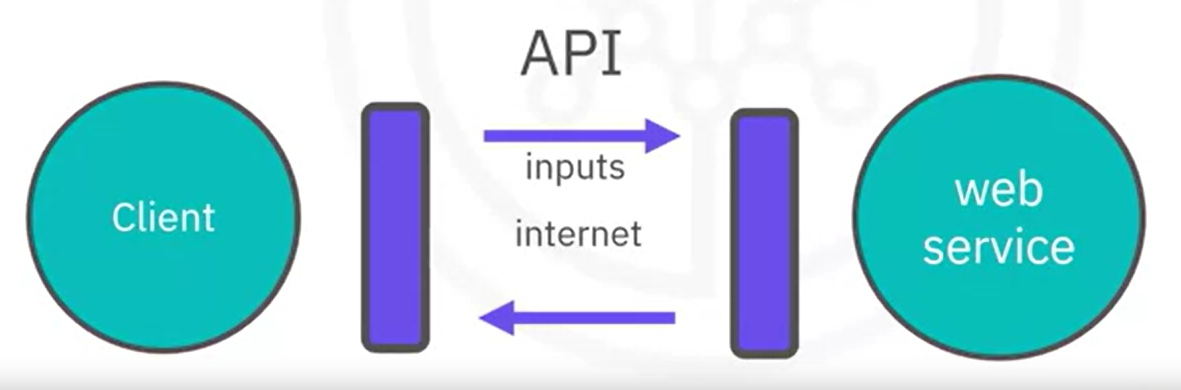
* It allows communication between two softwares



* For example, consider TensorFlow at backend (written in C++) which can use APIs for different languages such as Python, JS, C++, Java, Julia and Go.
* API is the part of the library you see while the library contains all components of the program

Rest APIs:

Representational State Transfer APIs (Rest APIs) allow to communicate through the internet following certain rules/conditions to use the resources.



Data sets:

Data set is a structured collection of data.

Types of data structures include:

* Tabular data – For example, CSV (Comma separated value) files which is also a delimited text file
* Hierarchical data – Data is organized in a tree format
* Network data – Data is organized in a graph format
* Raw files – For example, Images and audio files
* MNIST (Modified national Institute of standards and technology) data set is popular for data science – commonly used to train image processing systems

Data ownership:

* Private data
* Open data is fundamental to data science and machine learning. One such open data portals from around the world can be found in <http://datacatalogs.org/>
  + Governmental, intergovernmental and organization open portal websites include
    - <http://data.un.org/> (United Nations)
    - <http://www.data.gov/> (USA)
    - <http://www.europeandataportal.eu/en/> (Europe)
  + Kaggle – online data science community at <https://www.kaggle.com/datasets>
  + Google data set search – <https://datasetsearch.research.google.com/>

***Note:*** Community data license agreement is a must and easy way to share open data. For example, <http://cdla.io> (A Linux foundation project to use, modify and share data) includes CDLA-Sharing (use and modify data while publication only under same terms) and CDLA-Permissive (use and modify data with no obligations)

Open datasets:

**Government Data:**

* <https://www.data.gov/>
* <https://www.census.gov/data.html>
* <https://data.gov.uk/>
* <https://www.opendatanetwork.com/>
* <https://data.un.org/>

**Financial Data Sources:**

* <https://data.worldbank.org/>
* <https://www.globalfinancialdata.com/>
* <https://comtrade.un.org/>
* <https://www.nber.org/>
* <https://fred.stlouisfed.org/>

**Crime Data:**

* <https://www.fbi.gov/services/cjis/ucr>
* <https://www.icpsr.umich.edu/icpsrweb/content/NACJD/index.html>
* <https://www.drugabuse.gov/related-topics/trends-statistics>
* <https://www.unodc.org/unodc/en/data-and-analysis/>

**Health Data:**

* <https://www.who.int/gho/database/en/>
* <https://www.fda.gov/Food/default.htm>
* <https://seer.cancer.gov/faststats/selections.php?series=cancer>
* <https://www.opensciencedatacloud.org/>
* <https://pds.nasa.gov/>
* <https://earthdata.nasa.gov/>
* <https://www.sgim.org/communities/research/dataset-compendium/public-datasets-topic-grid>

**Academic and Business Data:**

* <https://scholar.google.com/>
* <https://nces.ed.gov/>
* <https://www.glassdoor.com/research/>
* <https://www.yelp.com/dataset>

**Other General Data:**

* <https://www.kaggle.com/datasets>
* <https://www.reddit.com/r/datasets/>

Proprietary datasets contain data primarily owned and controlled by specific individuals or organizations.

Data Asset eXchange (DAX) – IBM research data sets – <https://developer.ibm.com/>

**Machine Learning Models- Models to Predictions**

Types of ML models (or algorithms) include supervised, unsupervised and reinforcement.

Supervised Learning –

* Input is provided
* Comprises of 2 model types – Regression and Classification

Unsupervised Learning -

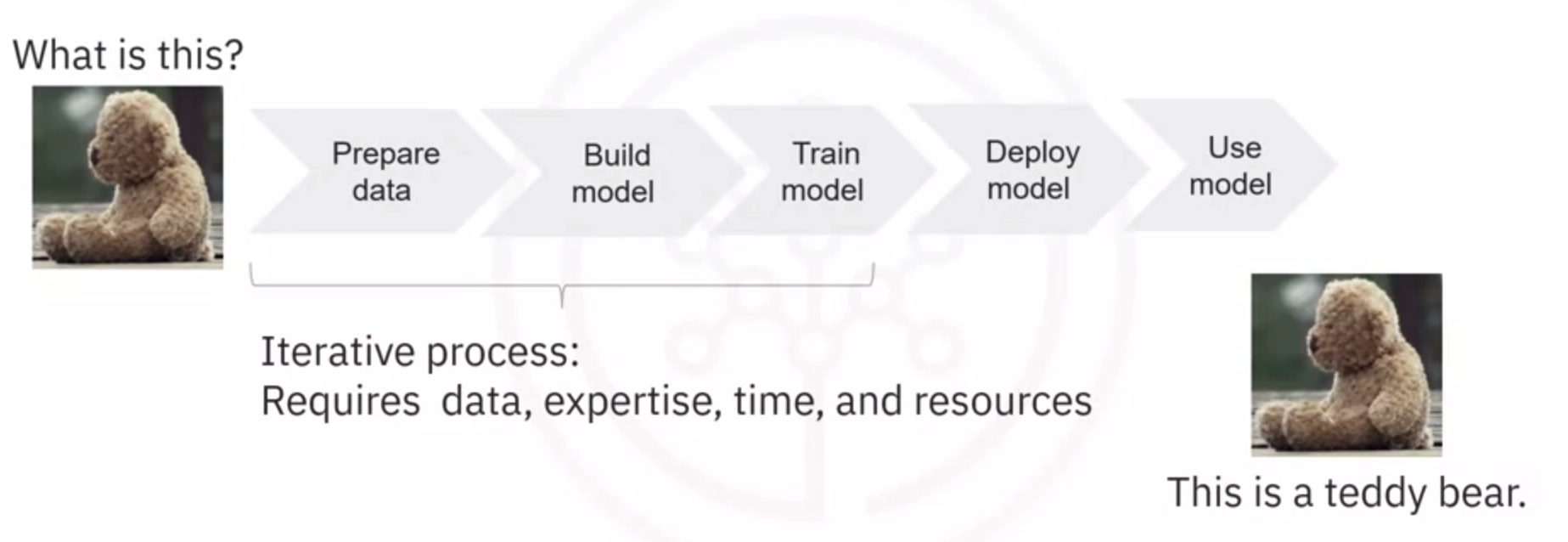
* Data is not labelled and no external help is provided for training the model
* Example is Clustering models

Reinforcement Learning –

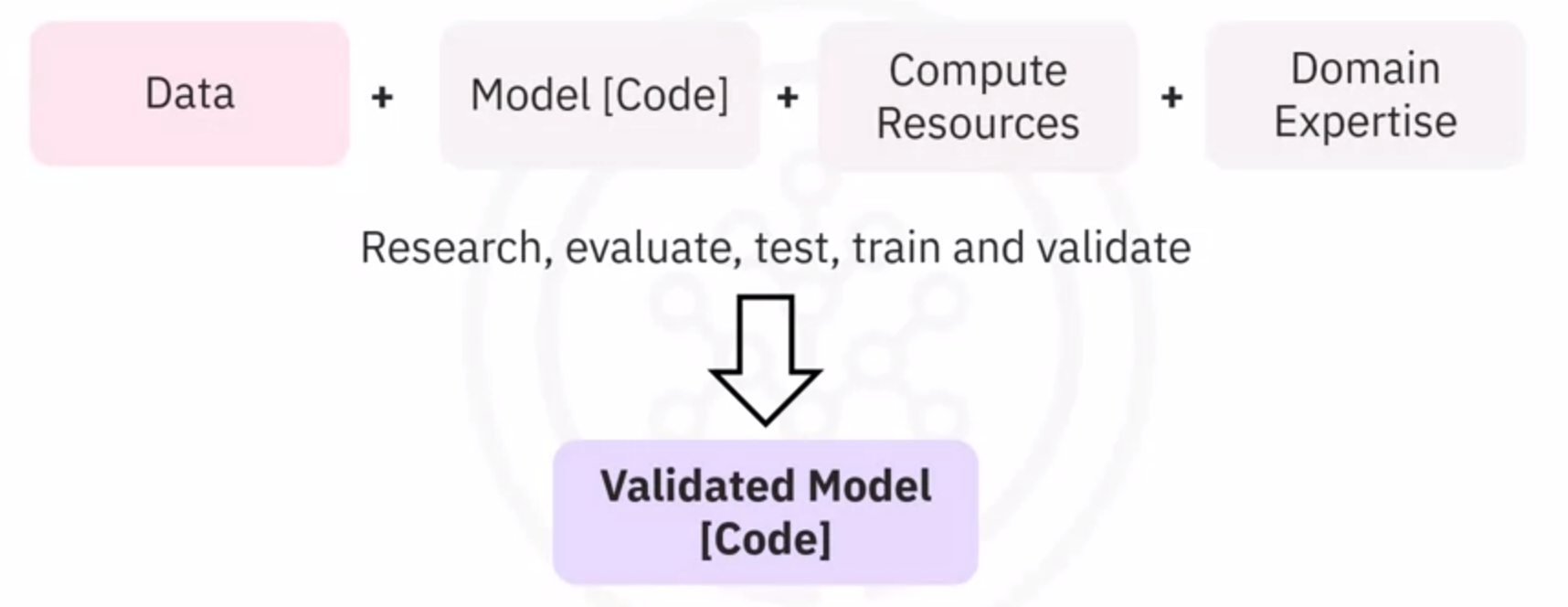
* Conceptually similar to human learning processes
* Examples include robot learning to walk, playing chess and video games

Deep Learning –

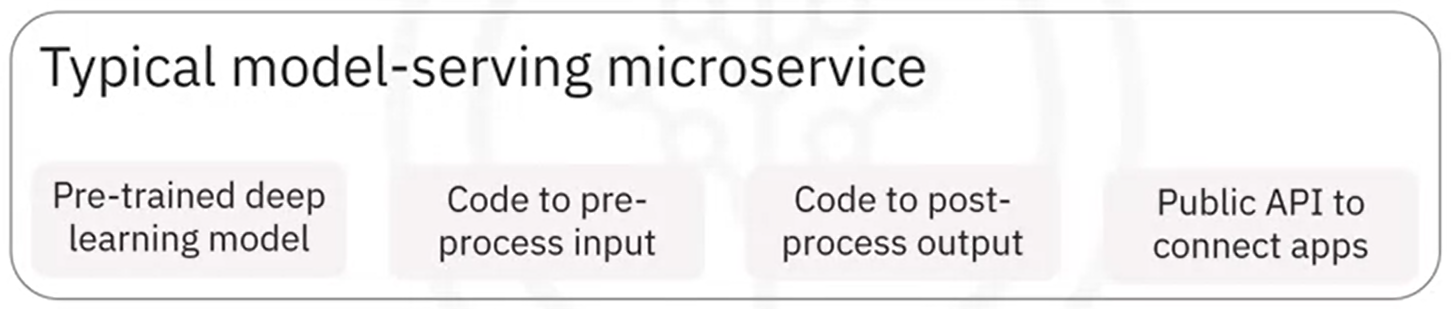
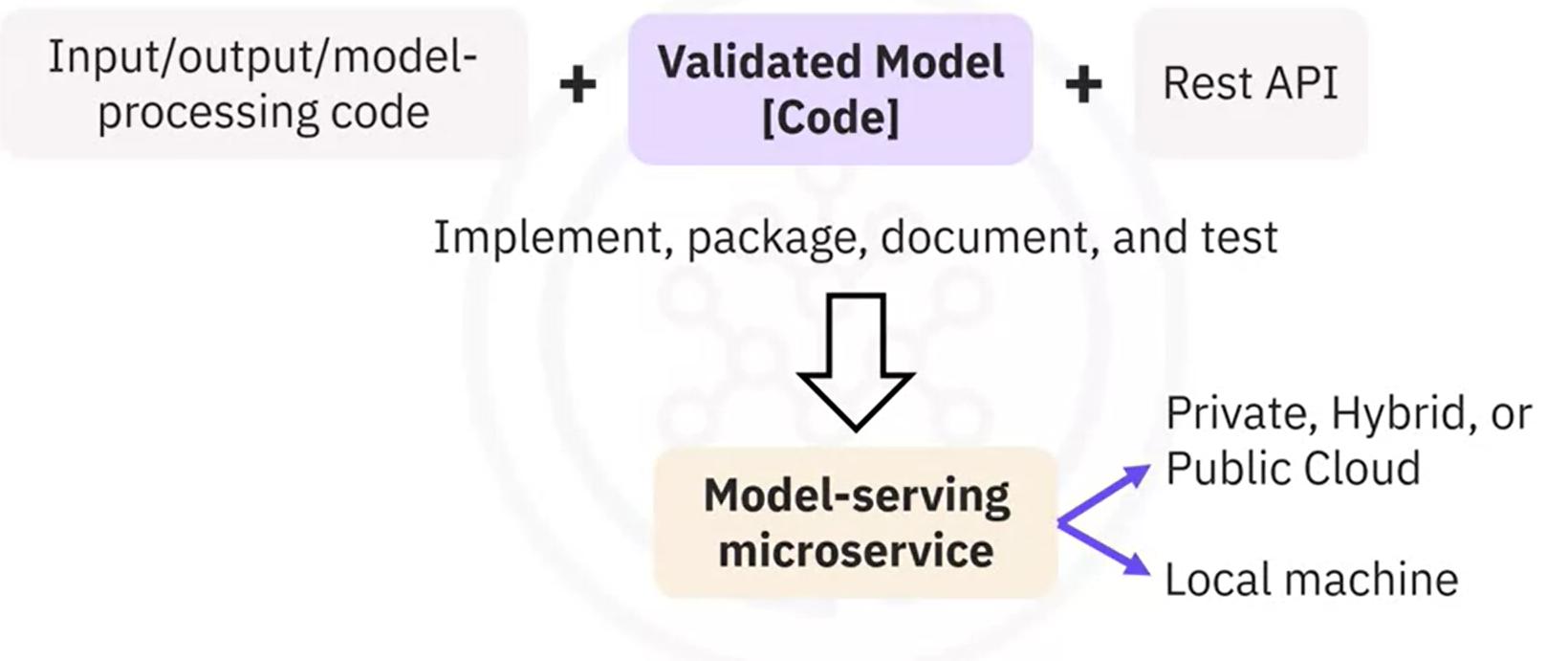
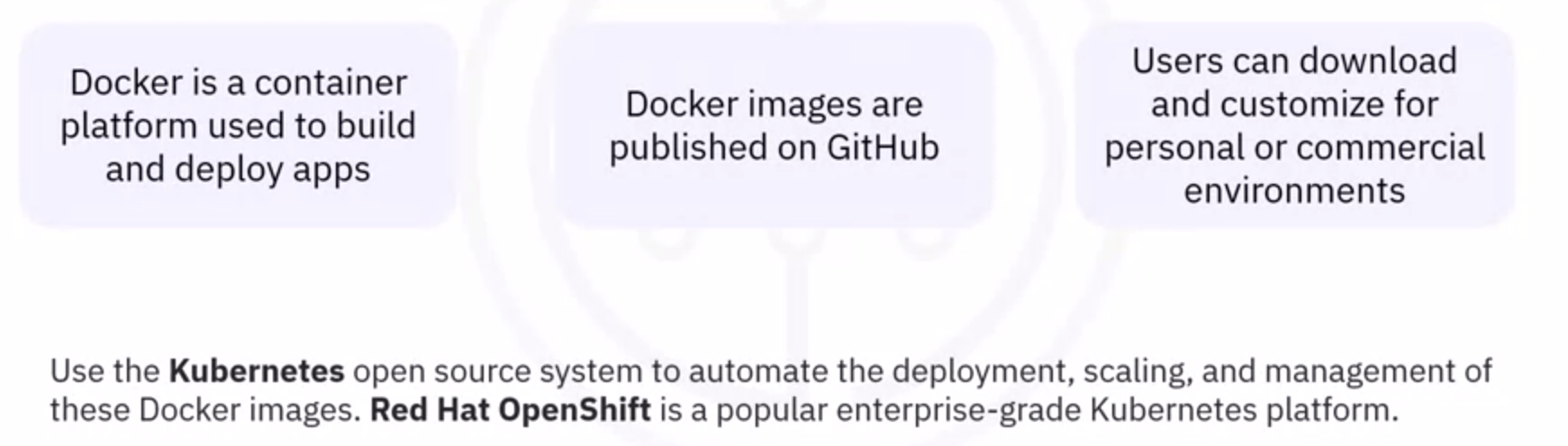
* Tries to loosely emulate human brain working
* Applications include Natural language processing, Analysis of image, audio and video, Time series analysis
* Requires large datasets of labelled data and compute intensive
* Require special purpose hardware
* DL models are implemented using frameworks such as TensorFlow, PyTorch, Keras. Most models provide a ‘model-zoo’ (model repository)
* Using models to solve a problem:



How models are created -



MAX (Model Asset eXchange) –

* Free open source repository for ready-to-use and customizable deep learning micro services
* Micro services are be pre-configured to use pre trained DL models which are then fully tested and deployed to solve common business problems.
* MAX includes models of variety of domains such as Object detection, Image, audio, video and text detection, Named entity recognition, Image to text translation, Human pose detection and more.
* 
* Model-serving micro-services creation –
* Max Model –serving micro services are built and distributed as open source Docker images
  + 
* Ml-exchange.org has multiple pre-defined models

**Week 4:**

**Jupyter Notebook and Jupyter Lab:**

Jupyter stands for Julia, Python and R. It however also supports Java, Scala, MATLAB, Octave, Ruby, JavaScript.

Jupyter notebook records data science experiments and results. It allows combining text, code blocks and code output into a single file. You can also export file into PDF or HTML file format.

Jupyter Lab is an open source application which allows access to multiple jupyter notebook files and data files. It enables working in integration manner and is compatible with several file formats.

Command: $pip install jupyterlab

**Jupyter Notebook:**

There are 2 types of cells:

* Markdown cell - Markdown cells allow us to apply several formatting options with the Markdown language – Headings, Links, Tables, Paragraphs, Bold text, Lists and more.
* Code cell

**Jupyter Kernel:**

It acts as a computational engine and executes the code in Notebook file. You can change to different kernel as per the programming language used to write code.

In JupyterLite, the default kernel is **Pyolite**. Similarly, if you are working on JupyterLab, the kernel will be **Python 3** or **Python 2**, depending on the version.

When the kernel is in execution, the state of the kernel is changed from **Idle** to the **Running** state.

**Markdown:**

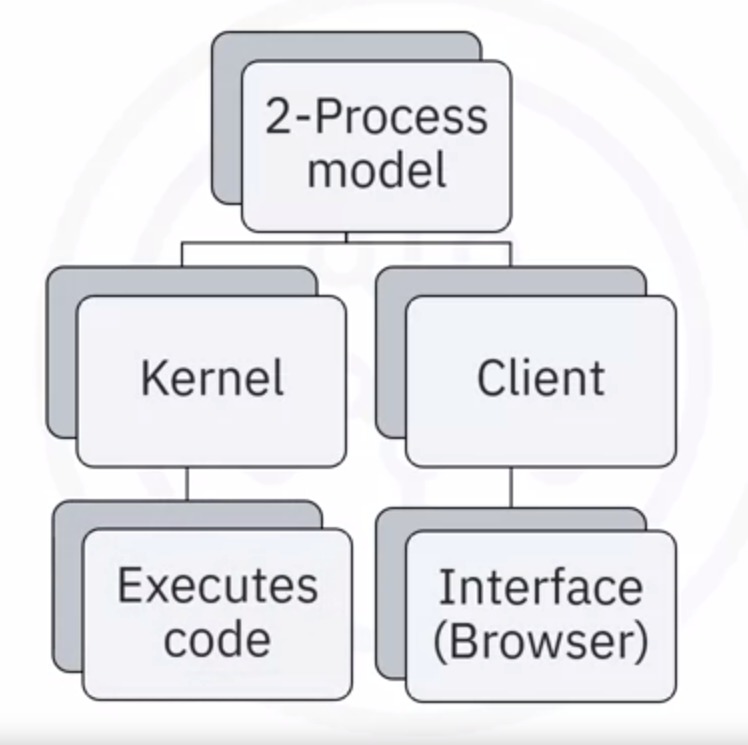
* 1. Headings – add # before a word or phrase. Increase 3 symbols for different level of headings.
  2. Bold text – add \*\* before and after a sentence or \_\_ before and after a sentence.
  3. Italic text – add \* or \_ before and after a sentence
  4. Bold and Italic text – add \*\*\* or \_\_\_ before and after a sentence
  5. Hyperlinks – Create hyperlink using the format [Name of the link](link url). For example, [Skills Network](<https://skills.network/>)

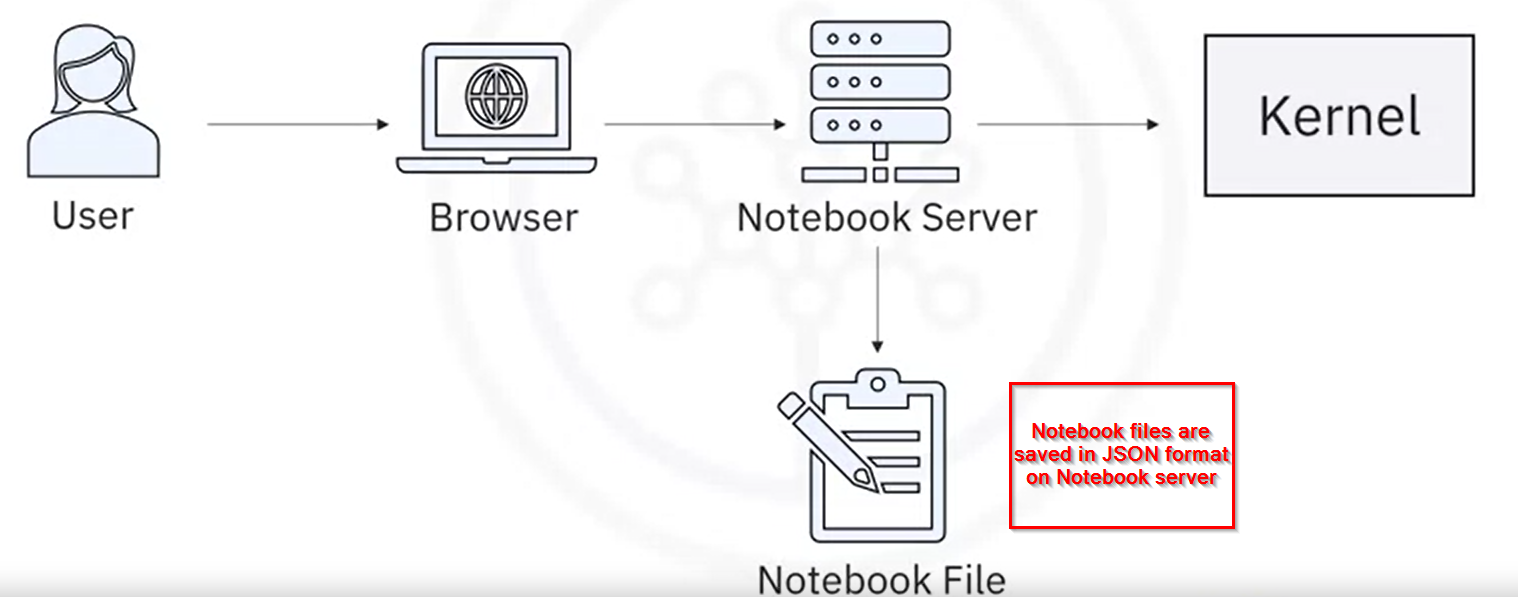
To display link without a name, put link in <>

* 1. Tables –
     + hyphens (----) for column headers
     + pipes | to separate each column
     + Text on a new line to separate each row
     + For example,
       - * | Country Name | Capital |
         * | -------------| ------ |
         * | United States | Washington DC |
         * | Australia | Canberra |
         * | India | New Delhi |
  2. Unordered lists – you can add \*, -, + in front of the items in the list
  3. Ordered lists – you can add numbers followed by periods before items.
     + For example,
       - 1. First item
       - 2. Second item

**Jupyter Architecture:**

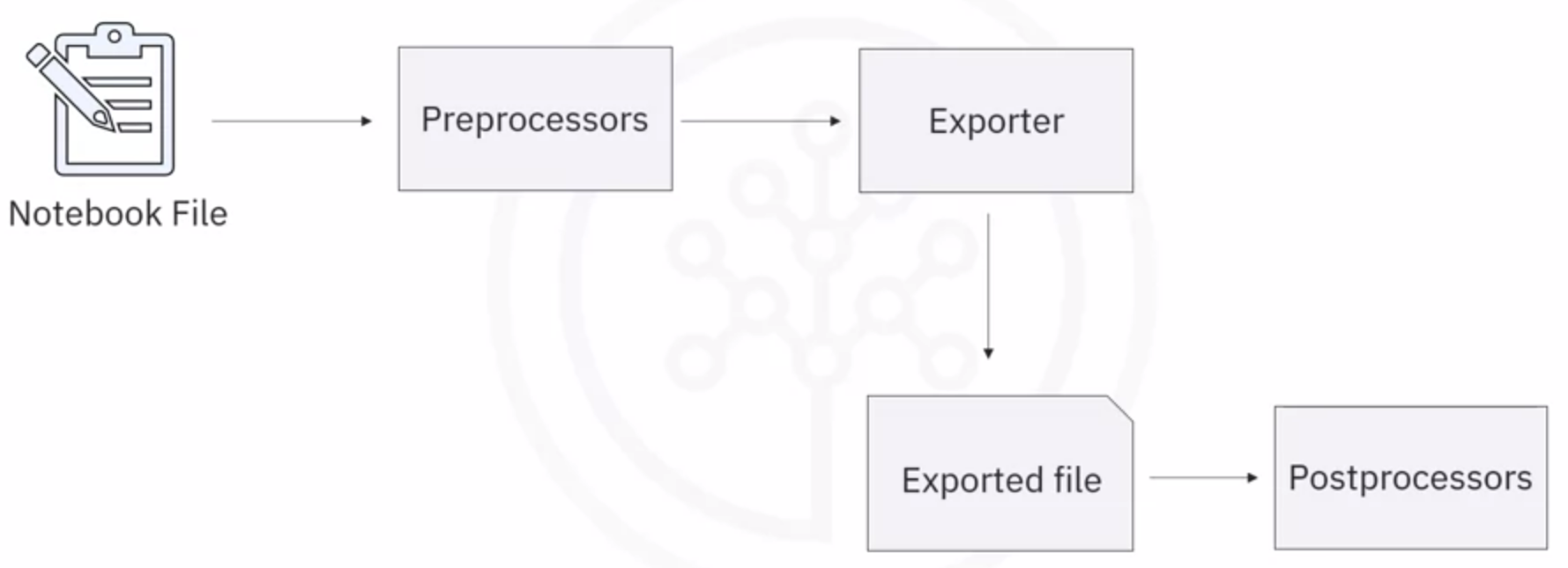
It involves a 2-process model as shown below





The Jupyter architecture uses NB convert tool to convert files to other formats.

For example, to convert a jupyter notebook into a HTML file the following process is used:



Jupyter notebook is a popular computational notebook because it supports dozens of programming languages. Jupyter Lab and VS Code are environments for creating and modifying jupyter notebooks on local device.

Jupyter Lab is an open source, web-based application based on Jupyter Notebook and it enables code creation, interactive visualization, text and equations. It includes pre installed python libraries.

Anaconda is an open source distributor for python and R (top languages used in data science and ML). It installs new packages without a CLI (command line interface). Anaconda navigator can launch multiple applications on local device.

VS Studio is a free, open source editor for debugging and task running operations. It is a popular development environment tool.

**Cloud based Jupyter environments:**

JupyterLite and Google Colaboratory are used to create and modify Jupyter Notebooks.

**JupyterLite –**

* Lightweight tool built from JupyterLab components that executes entirely in the browser.
* A dedicated Jupyter server is not required, only a web server is required to allow deploying JupyterLite as a static website.
* It supports many visualization libraries such as Altair, Plotly and ipywidgets.
* JupyterLite is a distributor of JupyterLab and includes latest JupyterLab improvements and features.
* Open browser and run <https://jupyter.org/try-jupyter/lab/> and slick Python(Pyodide)
* For cloud based jupyter environment, common kernels are Python Pyodide and Python Pyolite.

**Google Colaboratory or Google Colab -**

* A free jupyter notebook environment whioch runs entirely in the cloud
* Google Colab Jupyter notebooks execute on browsers while Google Colab projects are stored on Google drive and GitHub
* Projects can be cloned on GitHub and executed in Google Colab
* Most libraries are pre-installed

The Jupyter architecture uses the NB convert tool to convert files to other formats.

Popular source for jupyter notebooks: <https://github.com/jupyter/jupyter/wiki>

NB Viewer to view jupyter notebook files: <https://nbviewer.jupyter.org/>

**Jupyter notebooks for data science:**

* First, you start with exploratory data analysis, for which this notebook is highly recommended:

<https://nbviewer.jupyter.org/github/Tanu-N-Prabhu/Python/blob/master/Exploratory_data_Analysis.ipynb>

* For data integration/cleansing at a smaller scale, the python library\_pandas\_is often used. For this task, you can have a look at this notebook:

<https://towardsdatascience.com/data-cleaning-with-python-using-pandas-library-c6f4a68ea8eb>

* If you want to know more about clustering, have a look at this notebook:

<https://nbviewer.jupyter.org/github/temporaer/tutorial_ml_gkbionics/blob/master/2%20-%20KMeans.ipynb>

* And finally, if you want an in-depth notebook on the\_iris\_dataset, have a look at this:

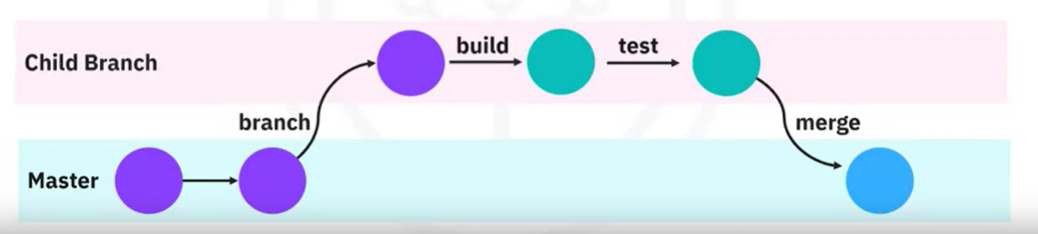
<https://www.kaggle.com/lalitharajesh/iris-dataset-exploratory-data-analysis>

**Week 5:**

**RStudio and GitHub:**

* R is a statistical programming language used for data processing and manipulation. It supports importing data from flat files (files stored in harddisk), databases, web, statistical software (SPSS and Strata).
* R is a great tool for visualisation
* RStudio is an integrated development environment (IDE) for R language development
* Popular R libraries for data science include dplyr (for data manipulation), stringr (for string manipulation), ggplot (for data visualisation) and caret (for ML)
* To install R - <https://cran.r-project.org/bin/windows/base/> and top install RStudio - <https://posit.co/download/rstudio-desktop/>
* R has various data visualisation packages used for data science as per requirement
  + Command to install packages install.package <package name>
  + Example packages are ggplot (histograms, bar charts, scatterplots), Plotly (web-based data visualisation), Lattice (complex, multi variable data sets), Leaflet (interactive plots)
* To install R packages : install.packages("GGally", repos = "https://cran.r-project.org", type = "source")

**GitHub:**

* Git Repository model:
  + A Distributed Version Control System (DVCS) keeps track of changes to code, regardless of where it is stored Tracks source code
  + Coordinates among programmers i.e; tracks changes and supports non-linear workflows
* Git is a distributed version control system
  + Tracks changes to content
  + Provides a central point for collaboration
* Git allows for centralized administration
  + Teams have controlled access scope
  + The main branch should always correspond to deployable code
* GitHub is an online hosting service for Git repositories
  + Hosted by subsidiary of Microsoft
  + Offers free, professional and enterprise accounts
  + Has more than 100 million repos
  + Repository or repo is data structure for storing documents including application source codes and it track and maintain version control
* GitLab is a DevOps platform delivered as a single application
  + Provides access to Git repos
  + Provides source code management
  + Allows developers to collaborate, work from a local copy, branch and merge code, streamline testing and delivering with CI/CD
* 
* Master branch contains finished, deployable version of the code
* Create new branches for changes to the code and use Pull requests to share code changes for review
* Copy of a repo is called Fork

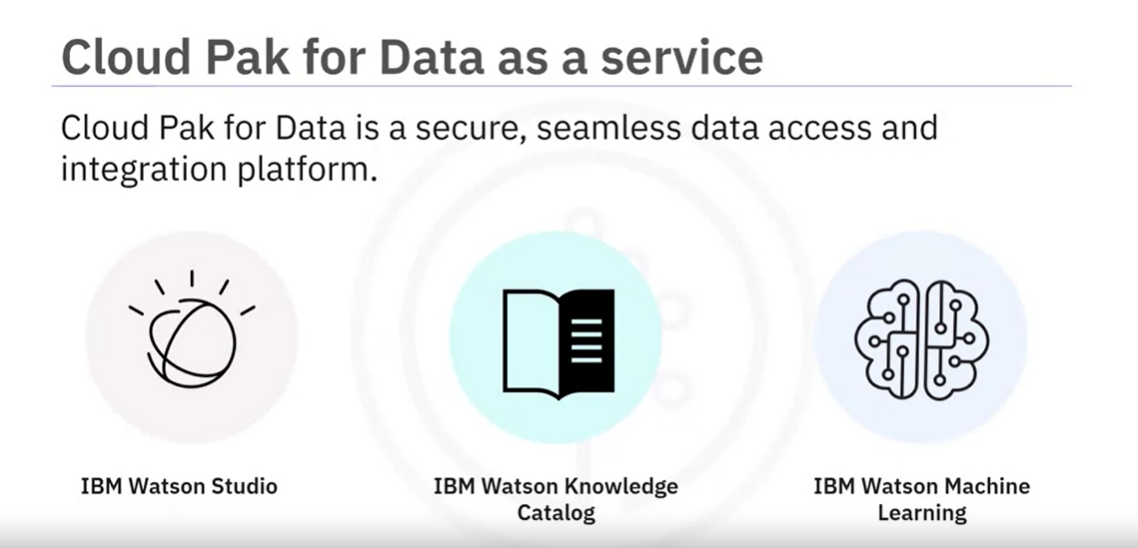
**Final exam questions:**

1. SQL is a database language
2. Regression model is used to predict a numerical value. For example, potential sales price of a used car
3. Open source tools sued for learning data science are RStudio and Jupyter Notebook/JupyterLite
4. RStudio does not provide function to store data in tables
5. Jupyter stands for Julia, Python and R
6. Watson Knowledge Catalog is used to discover ML assets in Watson studio
7. Data refinery helps build repeatable data pipelines for workloads of almost any size by creating a scheduled job and use custom environment to run the data flow/pipeline on different workloads
8. The node used to prepare the data for modelling in modeller flows is called ?
9. SPSS software package is used to specifically analyse data related to social sciences

**Week 7:**

Watson Studio is a collaborative platform for data science community present in IBM cloud

Community includes data analysts, data scientists, data engineers, developers and data stewards.



* Learning about development environments
* Data Science tools for data migration, data integration, data visualization model deployment and model monitoring
* Commercial and Cloud based tools for data asset management
* Language selection for data science including R, Java and Python
* Data Science packages, APIs, datasets and models such as machine learning and deep learning models for predictions (Model Asset eXchange on Watson Studio)
* Cloud based environment such as Jupyter Lite and use of GitLab