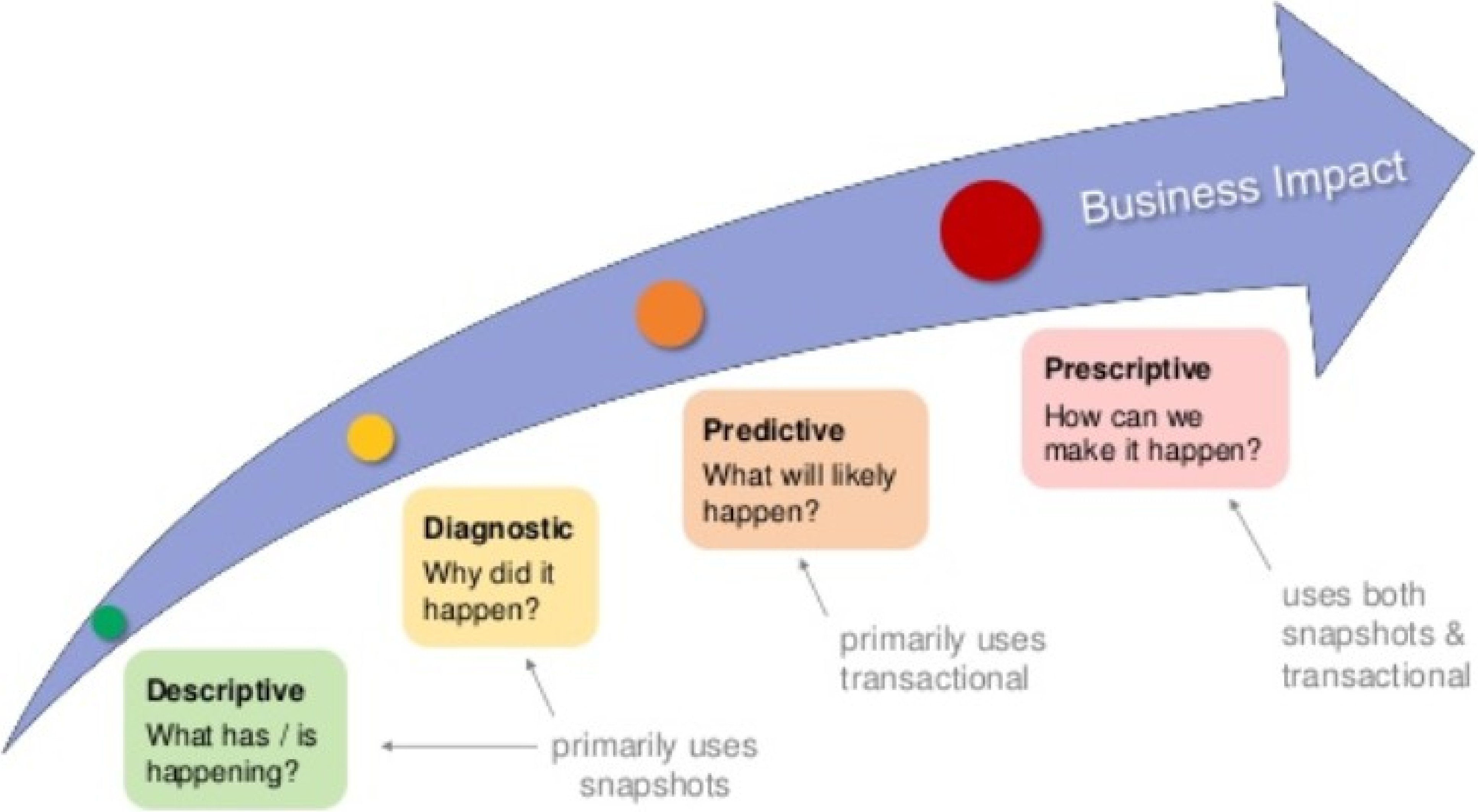
Data Analytics Practitioner

Overall Objective: Completion of Data Analytics Practitioner courses should equipped participants with the knowledge to be a Data Scientist.

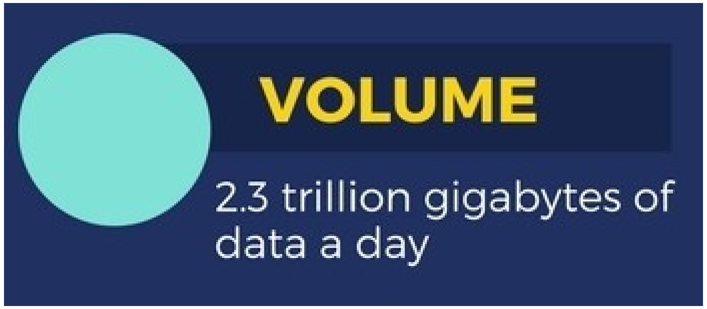
Core Objectives:

- Data Analytics Practitioner 1: Classroom based training that allows participants to learn programming language such as Python
- Data Analytics Practitioner 2: Similar to Data Analytics 2 courses, this course provides hands-on approach by using Machine Learning technique. Ideally, participants should work on a use case that is relevant to their job scope
- Data Analytics Practitioner 3: Consultation session to help kick start client’s analytics project

Current Analytics Landscape:

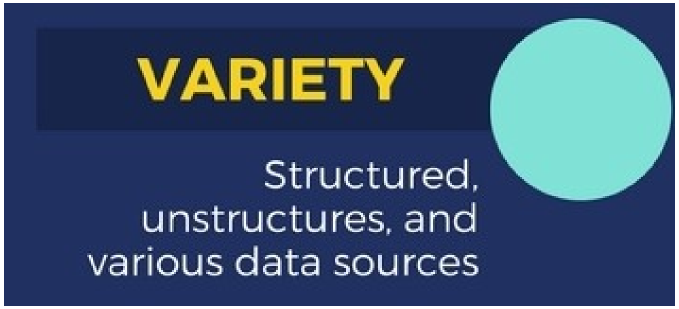


The 4Vs of Analytics:



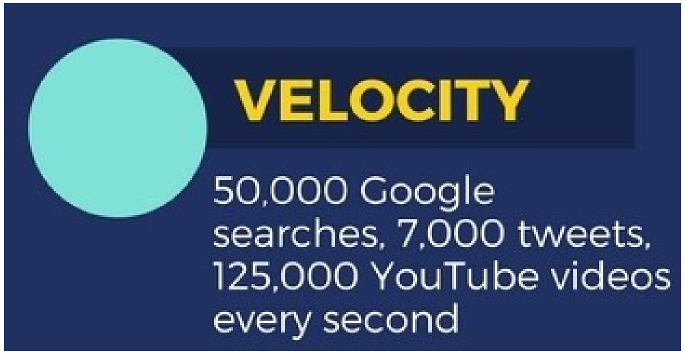
VOLUME

2.3 trillion gigabytes of data a day



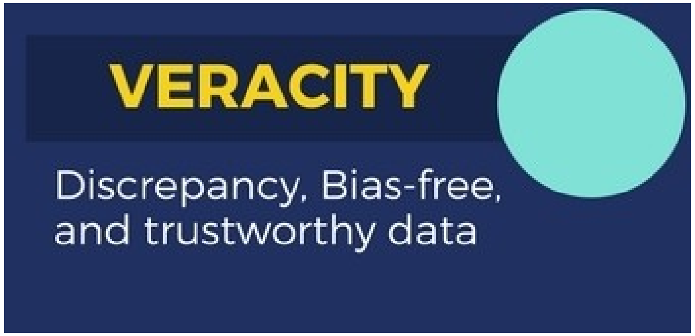
VARIETY

Structured, unstructures, and various data sources



VELOCITY

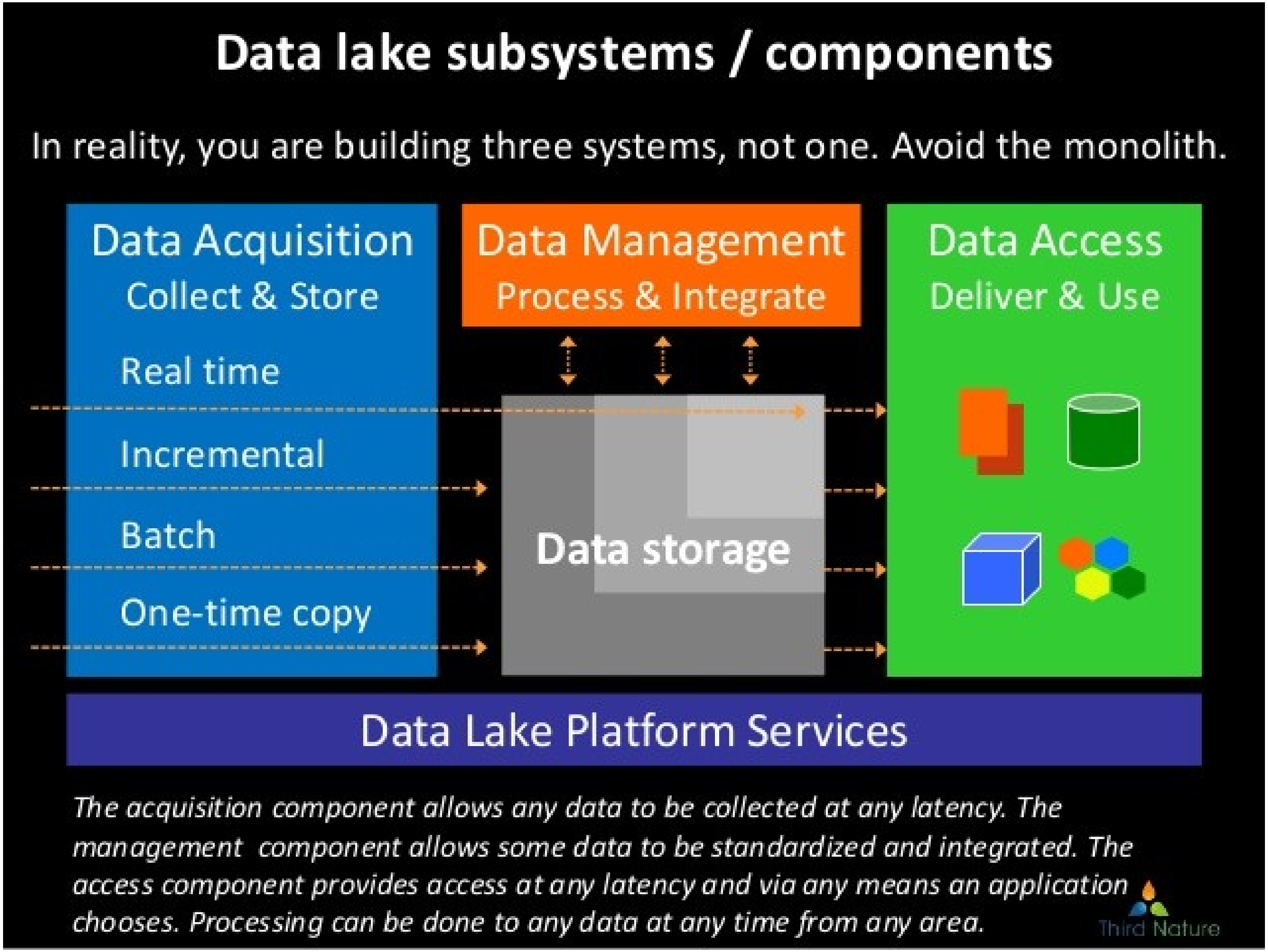
50,000 Google searches, 7,000 tweets, 125,000 YouTube videos every second



VERACITY

Discrepancy, Bias-free, and trustworthy data

Building an Analytics Datalake



Data lake subsystems / components

In reality, you are building three systems, not one. Avoid the monolith.

Data Acquisition
Collect & Store

Real time

Incremental

Batch

One-time copy

Data Management
Process & Integrate

Data storage

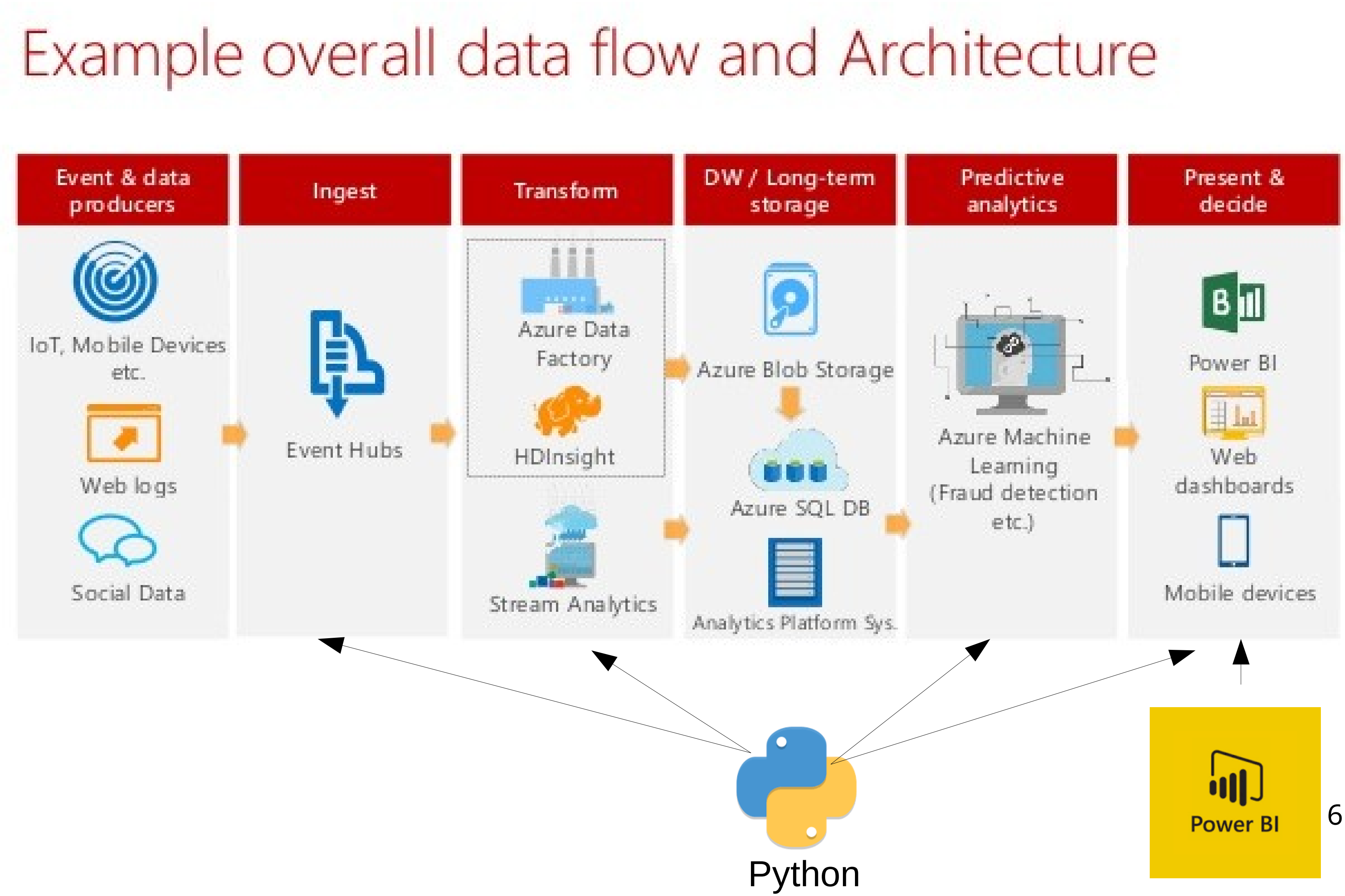
Data Access
Deliver & Use

Data Lake Platform Services

The acquisition component allows any data to be collected at any latency. The management component allows some data to be standardized and integrated. The access component provides access at any latency and via any means an application chooses. Processing can be done to any data at any time from any area.

Third Nature

Modern Data Analytics Platform:



Example overall data flow and Architecture

Event & data producers

IoT, Mobile Devices etc.

Web logs

Social Data

Ingest

Event Hubs

Transform

Azure Data Factory

HDInsight

Stream Analytics

DW / Long-term storage

Azure Blob Storage

Azure SQL DB

Analytics Platform Sys.

Predictive analytics

Azure Machine Learning (Fraud detection etc.)

Present & decide

Power BI

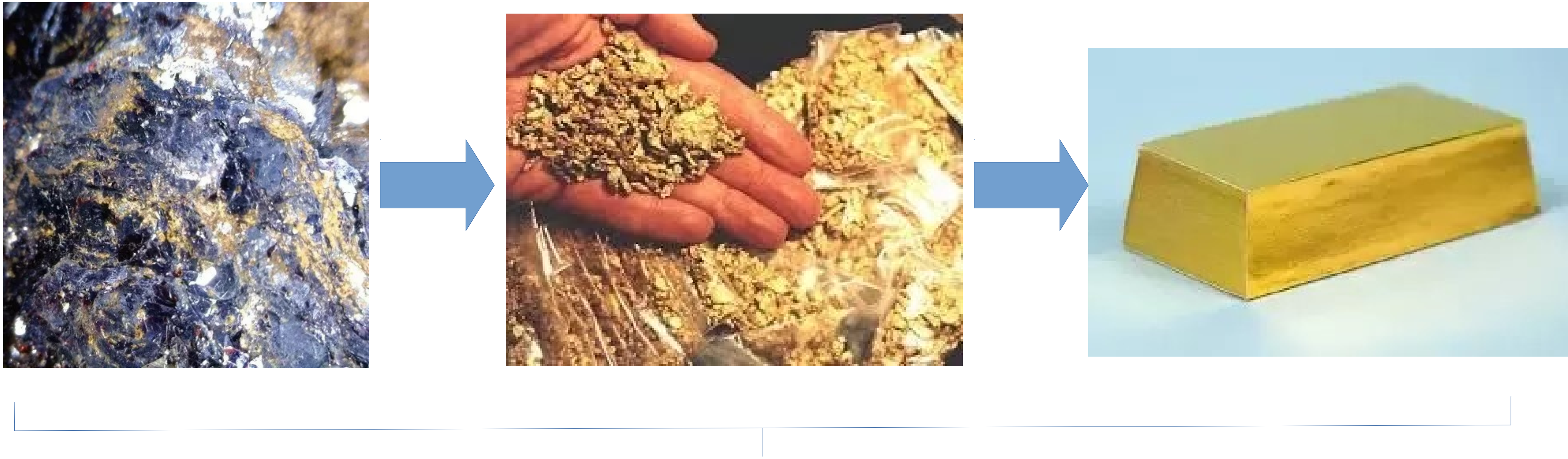
Web dashboards

Mobile devices

Power BI

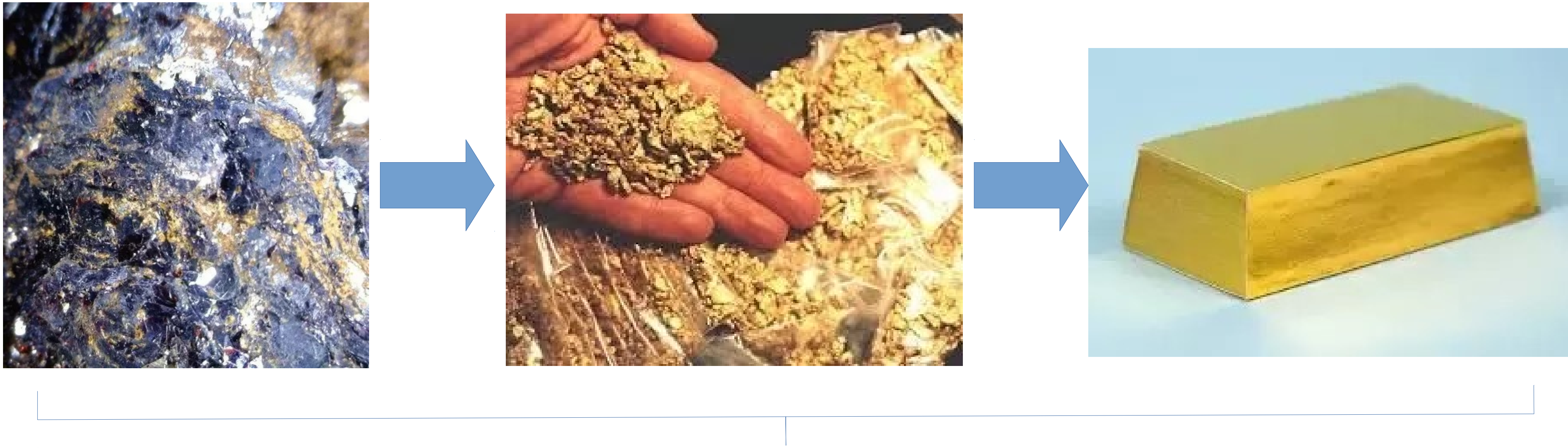
Python

Raw to Processed to Information



Raw data is the data that is measured and collected directly from machine, web, etc. ... The **processed data** is the type of data that is processed from raw data. Usually some kind of cleaning, transformation are performed to convert the raw data into a format that can be analyzed, visualized

Raw to Processed to Information



Personnel – data engineer, data scientist, data architect

Infra – database server, data warehouse, hadoop etc

Data management – SSIS, informatica, data quality etc

Predictive Analytics – python, R, SPSS etc

Business Intelligence – Powerbi, tableau, Qlik

Raw to Processed to Information

Personnel – data engineer, data scientist, data architect, **problem owner, marketing department, C-level, IT Department**

Infra – database server, data warehouse, hadoop etc

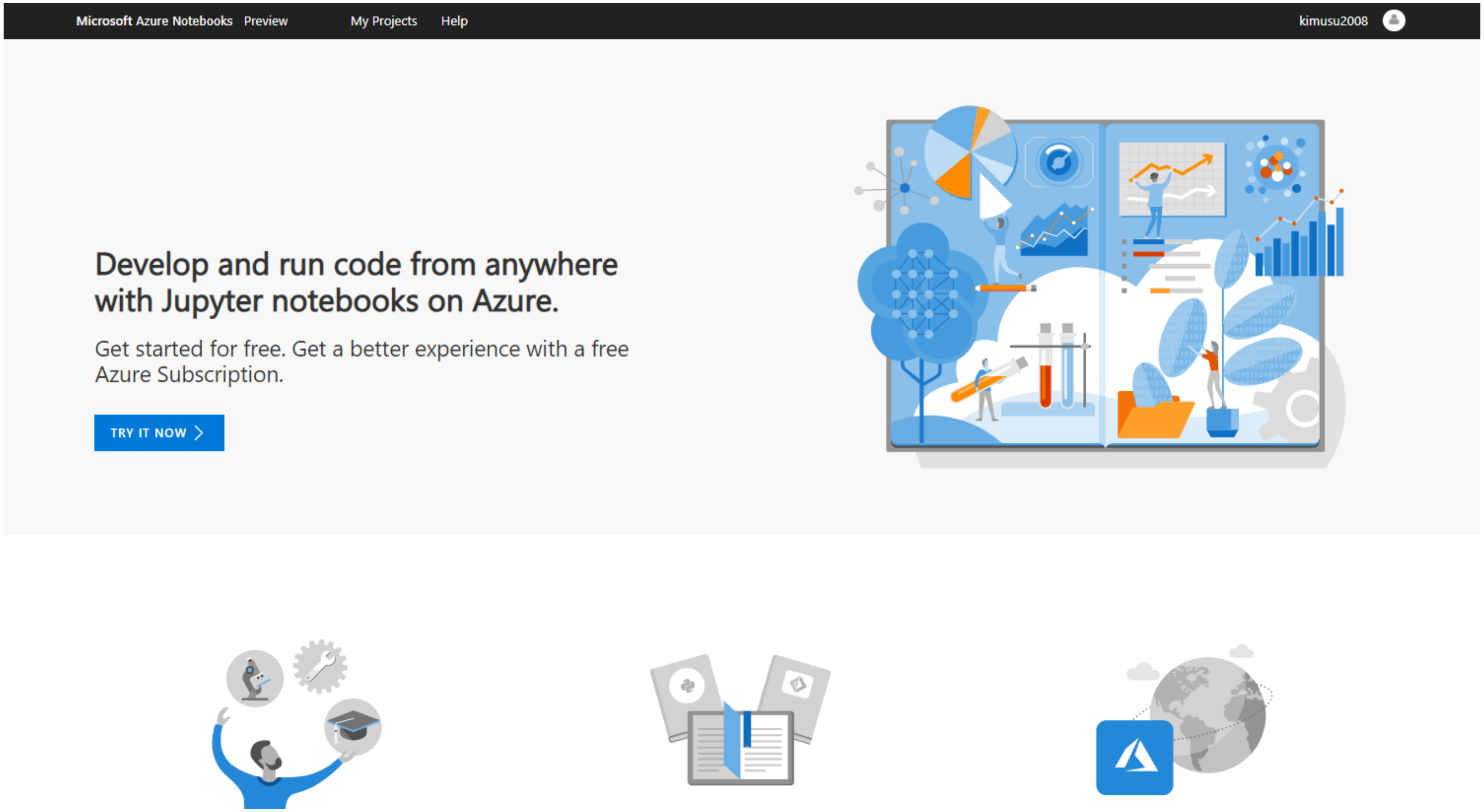
Data management – SSIS, informatica, data quality etc

Predictive Analytics – python, R, SPSS etc

Business Intelligence – Powerbi, tableau, Qlik

Data Analytics Practitioner 4: Classroom based training that allows participants to learn programming language such as Python

- Preview:
  - Using Azure python notebook (free online)

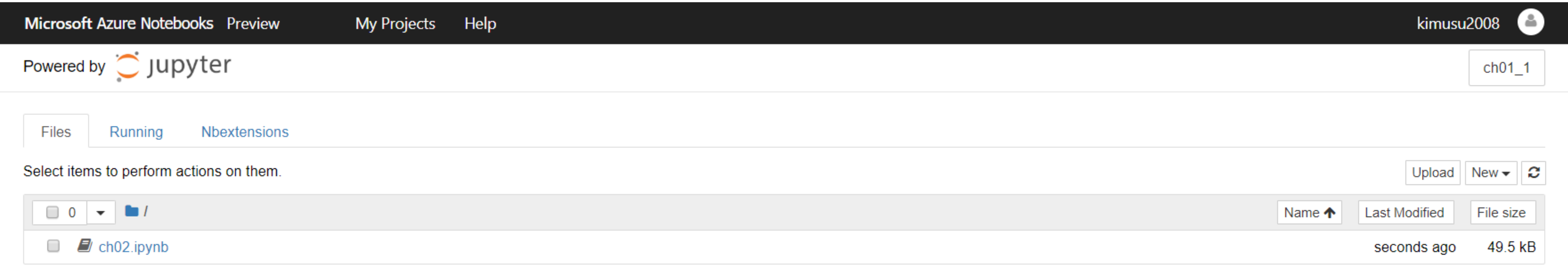


Develop and run code from anywhere with Jupyter notebooks on Azure.

Get started for free. Get a better experience with a free Azure Subscription.

Data Analytics Practitioner 4: Classroom based training that allows participants to learn programming language such as Python

- Preview:
  - Using Azure python notebook (free online)

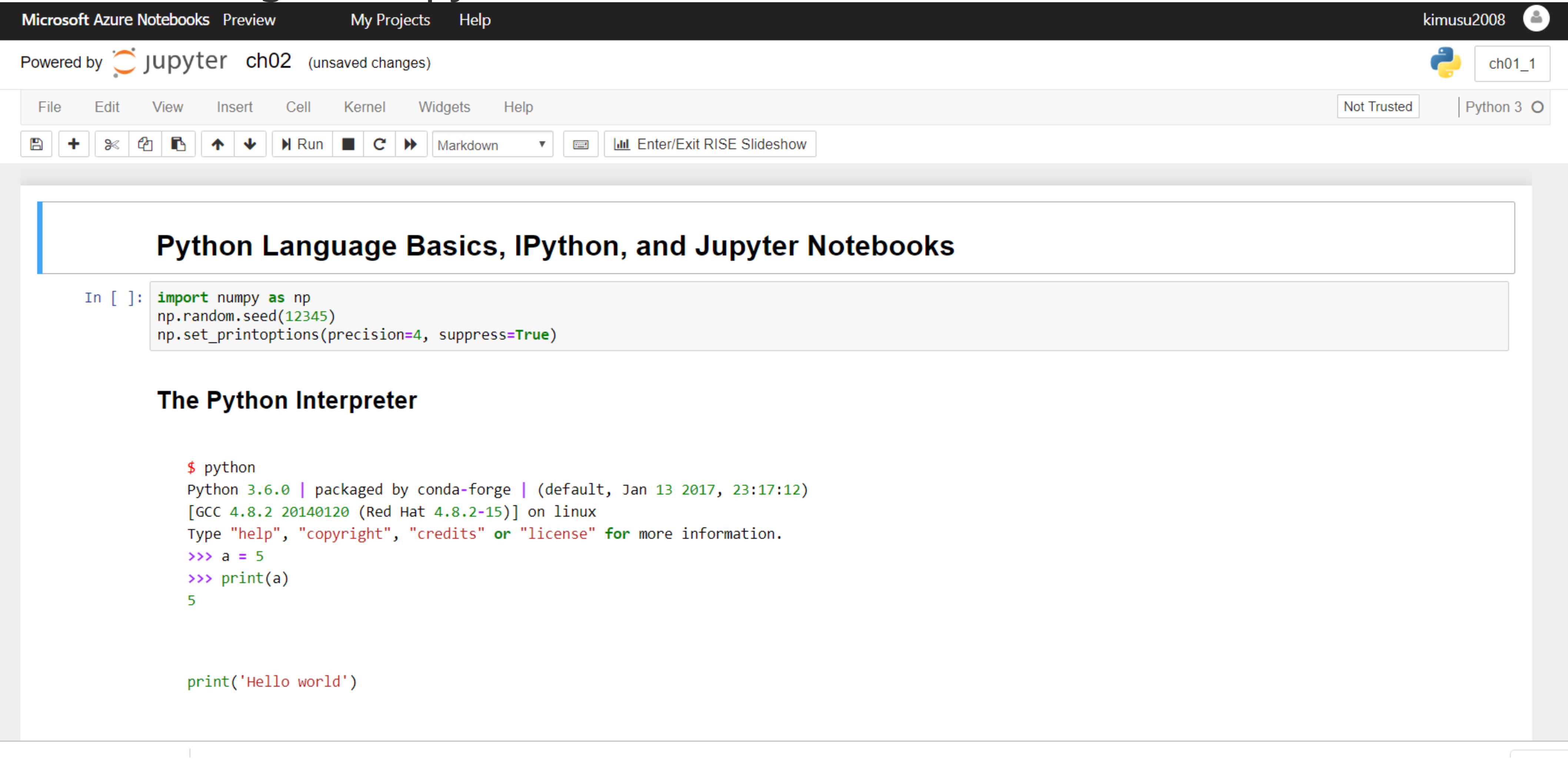


Select items to perform actions on them.

Name	Last Modified	File size
ch02.ipynb	seconds ago	49.5 kB

Data Analytics Practitioner 4: Classroom based training that allows participants to learn programming language such as Python

- Preview:
  - Using Azure python notebook (free online)



Python Language Basics, IPython, and Jupyter Notebooks

```
In [ ]: import numpy as np
        np.random.seed(12345)
        np.set_printoptions(precision=4, suppress=True)
```

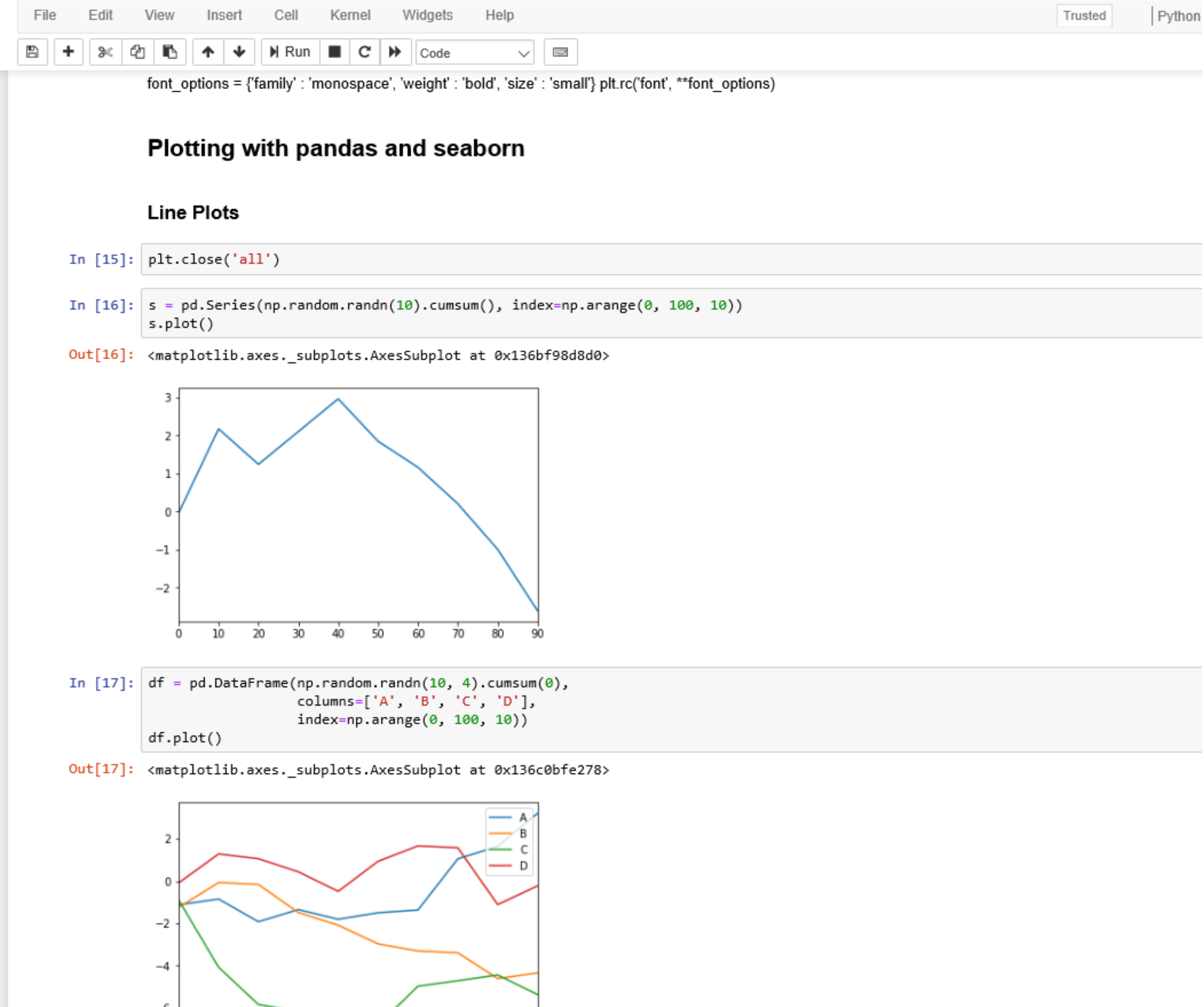
The Python Interpreter

```
$ python
Python 3.6.0 | packaged by conda-forge | (default, Jan 13 2017, 23:17:12)
[GCC 4.8.2 20140120 (Red Hat 4.8.2-15)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 5
>>> print(a)
5
```

```
print('Hello world')
```

Data Analytics Practitioner 4: Classroom based training that allows participants to learn programming language such as Python

- Preview:
  - Using Azure python notebook (free online)



font_options = {'family' : 'monospace', 'weight' : 'bold', 'size' : 'small'} plt.rc('font', **font_options)

Plotting with pandas and seaborn

Line Plots

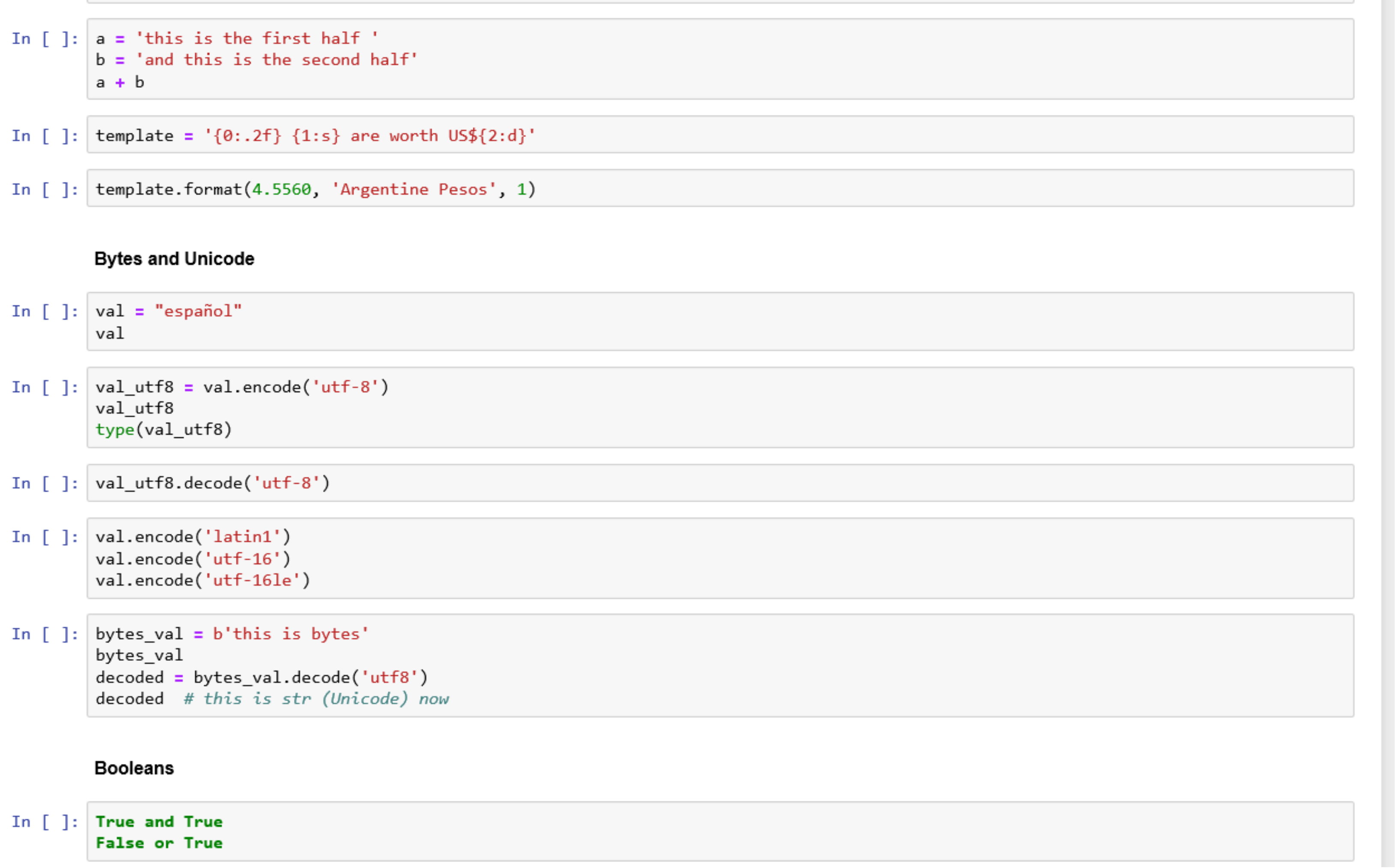
```
In [15]: plt.close('all')
```

```
In [16]: s = pd.Series(np.random.randn(10).cumsum(), index=np.arange(0, 100, 10))
         s.plot()

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x136bf98d8d0>
```

```
In [17]: df = pd.DataFrame(np.random.randn(10, 4).cumsum(0),
                           columns=['A', 'B', 'C', 'D'],
                           index=np.arange(0, 100, 10))
         df.plot()

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x136c0bfe278>
```

Data Analytics Practitioner 4: Classroom based training that allows participants to learn programming language such as Python

- Preview:
  - Using Azure python notebook (free online)



```
In [ ]: a = 'this is the first half '
        b = 'and this is the second half'
        a + b
```

```
In [ ]: template = '{0:.2f} {1:s} are worth US${2:d}'
```

```
In [ ]: template.format(4.5560, 'Argentine Pesos', 1)
```

Bytes and Unicode

```
In [ ]: val = "español"
        val
```

```
In [ ]: val_utf8 = val.encode('utf-8')
        val_utf8
        type(val_utf8)
```

```
In [ ]: val_utf8.decode('utf-8')
```

```
In [ ]: val.encode('latin1')
        val.encode('utf-16')
        val.encode('utf-16le')
```

```
In [ ]: bytes_val = b'this is bytes'
        bytes_val
        decoded = bytes_val.decode('utf8')
        decoded  # this is str (Unicode) now
```

Booleans

```
In [ ]: True and True
        False or True
```

Data Analytics Practitioner 4: Classroom based training that allows participants to learn programming language such as Python

- Preview:
  - The Pandas module is used for working with tabular data. It allows us to work with data in table form, such as in CSV or SQL database formats. We can also create tables of our own, and edit or add columns or rows to tables. Pandas provides us with some powerful objects like DataFrames and Series which are very useful for working with and analyzing data.
  - The Numpy module is mainly used for working with numerical data. It provides us with a powerful object known as an Array. With Arrays, we can perform mathematical operations on multiple values in the Arrays at the same time, and also perform operations between different Arrays, similar to matrix operations.
  - Last, but not least, the Matplotlib module is used for data visualization. It provides functionality for us to draw charts and graphs, so that we can better understand and present the data visually.

Data Analytics Practitioner 4: Classroom based training that allows participants to learn programming language such as Python

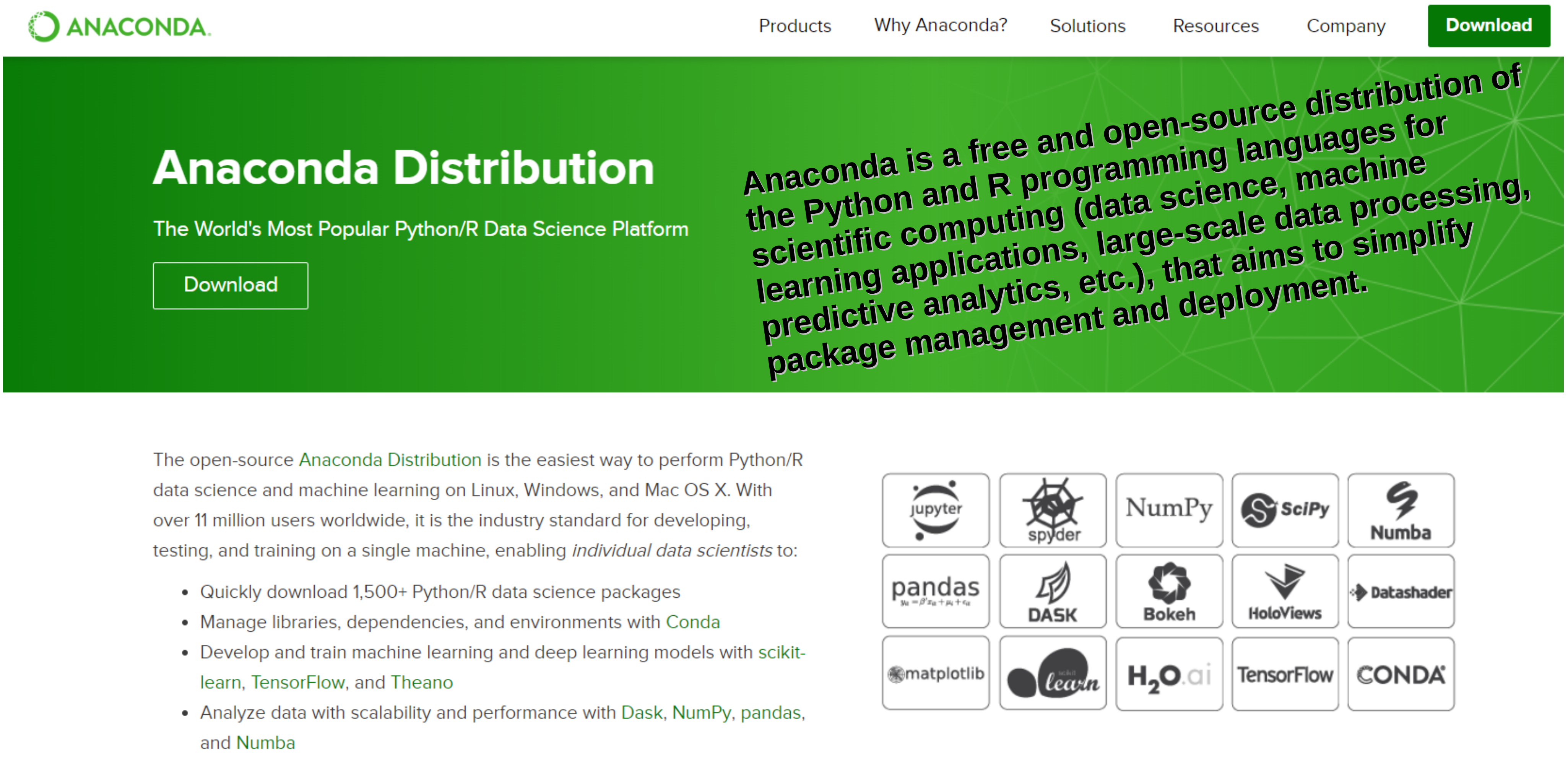
- Preview:
  - Pandas provides high level data manipulation tools built on top of NumPy. NumPy by itself is a fairly low-level tool. Pandas on the other hand provides rich time series functionality, data alignment, NA-friendly statistics, groupby, merge and join methods, and lots of other conveniences. It has become very popular in recent years in financial applications.
  - SciPy is a free and open-source Python library used for scientific computing and technical computing. SciPy contains modules for 1. optimization, 2. linear algebra, 3.integration, 4. interpolation, 5. special functions, 6. FFT, 7. signal and image processing, 8. ODE solvers and other tasks common in science and engineering

Data Analytics Practitioner 4: Classroom based training that allows participants to learn programming language such as Python

- Preview:
  - Python Language Basics and Jupyter Notebooks
  - Built-in Data Structures, Functions, and Files
  - NumPy Basics: Arrays and Vectorized Computation
  - Getting Started with pandas
  - Data Loading, Storage, and File Formats
  - Data Cleaning and Preparation
  - Data Wrangling: Join, Combine, and Reshape
  - Plotting and Visualization
  - etc

Data Analytics Practitioner 4: Classroom based training that allows participants to learn programming language such as Python

- Preview:
  - Using Anaconda Python Distribution (free download)



Anaconda Distribution

The World's Most Popular Python/R Data Science Platform

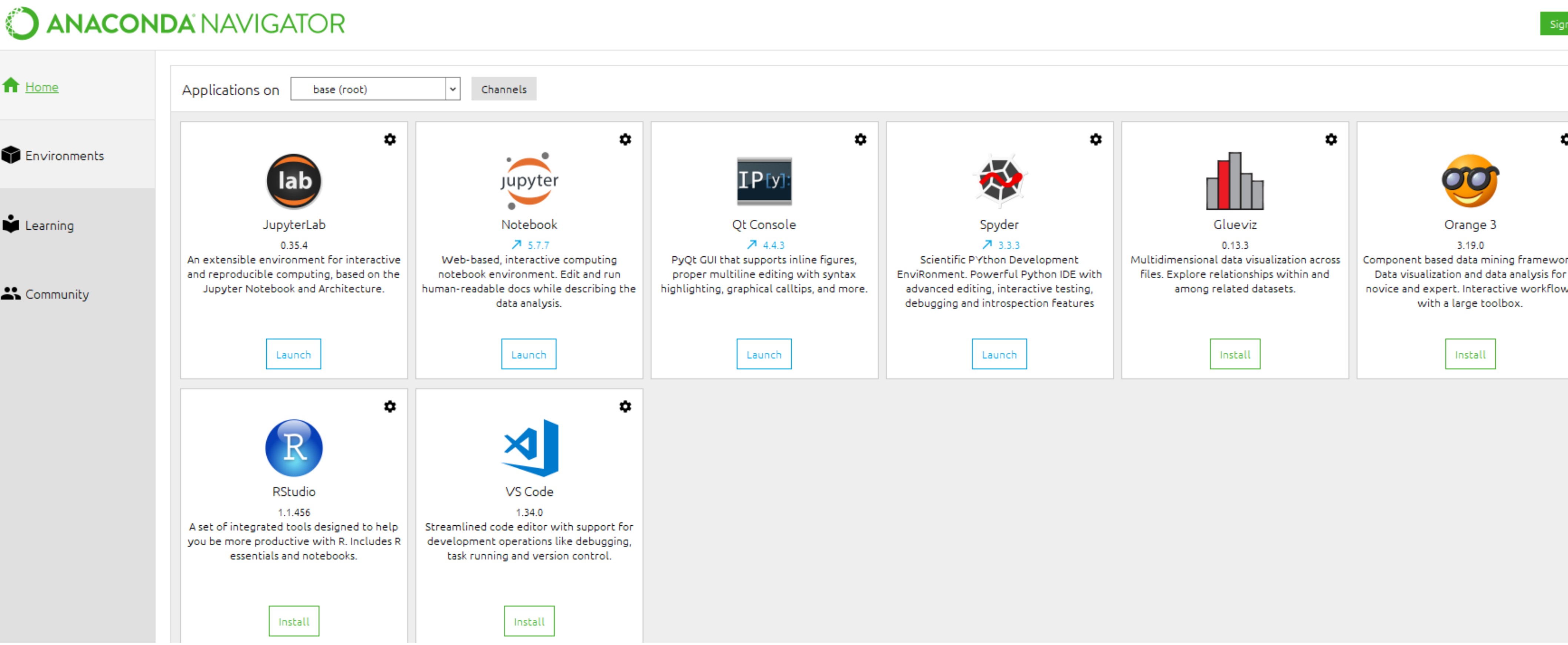
Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.

The open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 11 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download 1,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with Conda
- Develop and train machine learning and deep learning models with scikit-learn, TensorFlow, and Theano
- Analyze data with scalability and performance with Dask, NumPy, pandas, and Numba

Data Analytics Practitioner 4: Classroom based training that allows participants to learn programming language such as Python

- Preview:
  - Using Anaconda Python Distribution (free download)



JupyterLab
0.35.4
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
5.7.7
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Qt Console
4.4.3
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder
3.3.3
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Glueviz
0.13.3
Multidimensional data visualization across files. Explore relationships within and among related datasets.

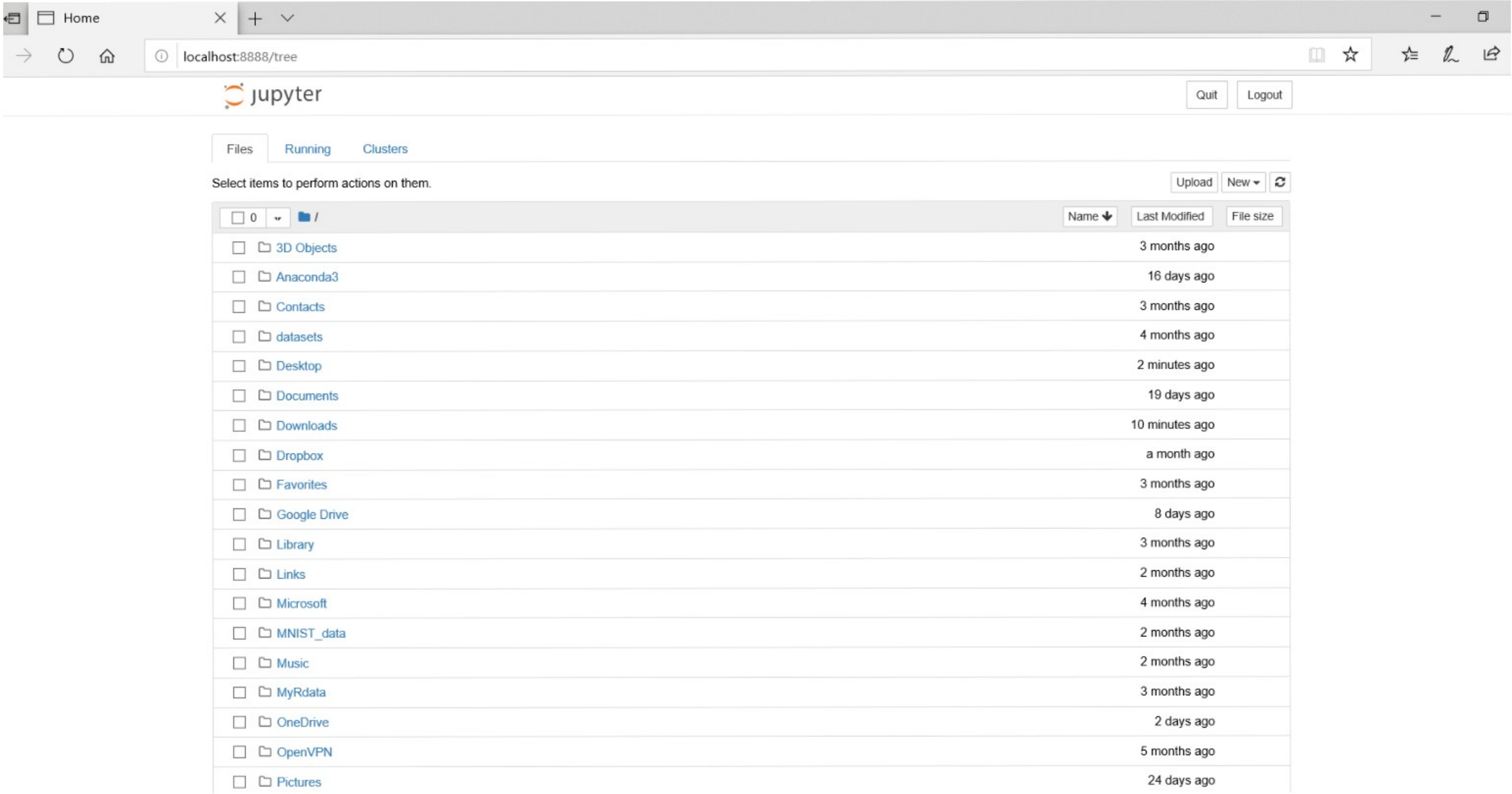
Orange 3
3.19.0
Component based data mining framewor Data visualization and data analysis for novice and expert. Interactive workflow with a large toolbox.

RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

VS Code
1.34.0
Streamlined code editor with support for development operations like debugging, task running and version control.

Data Analytics Practitioner 4: Classroom based training that allows participants to learn programming language such as Python

- Preview:
  - Using Anaconda Python Distribution (free download)

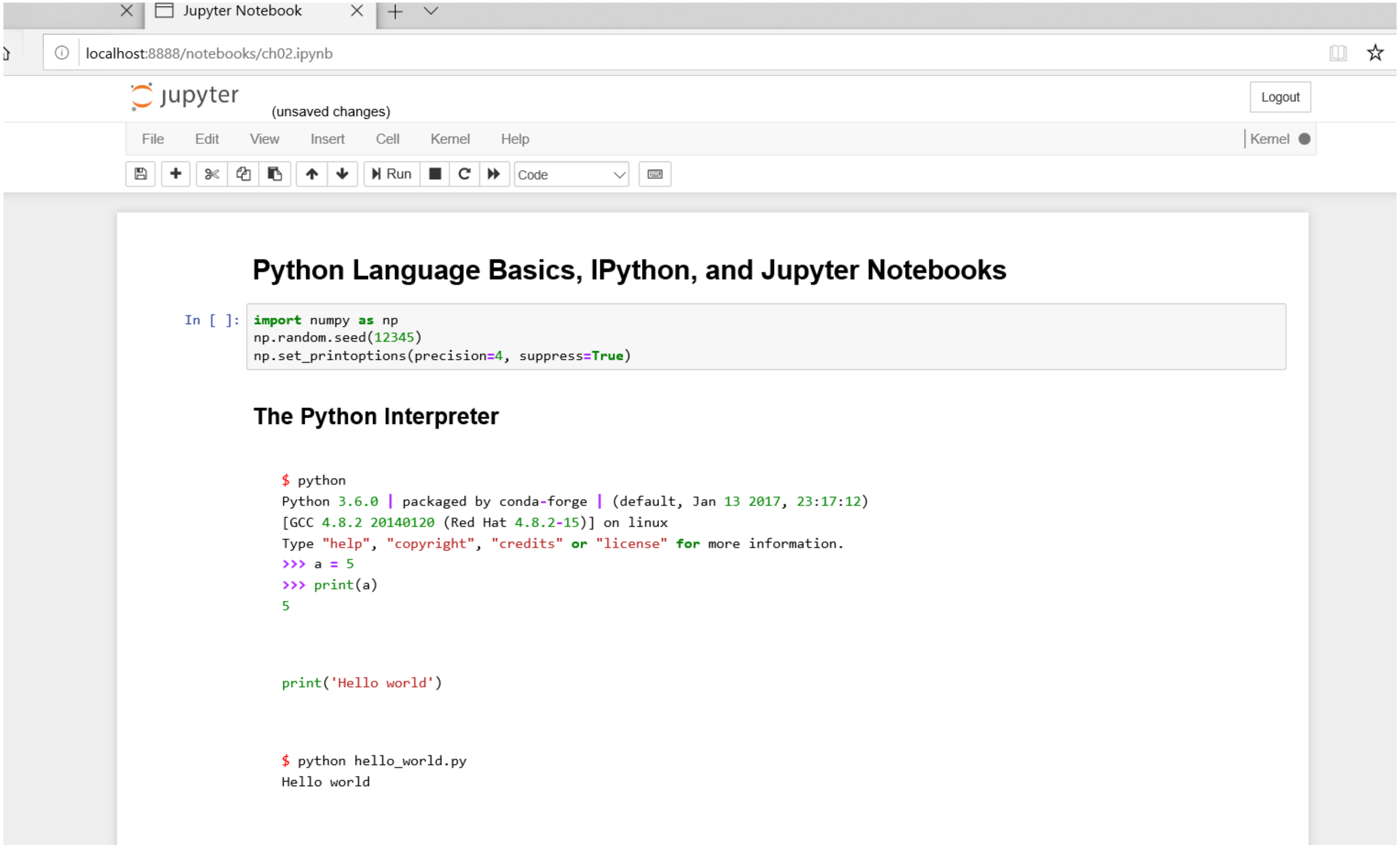


Select items to perform actions on them.

Name	Last Modified	File size
□ 3D Objects	3 months ago	
□ Anaconda3	16 days ago	
□ Contacts	3 months ago	
□ datasets	4 months ago	
□ Desktop	2 minutes ago	
□ Documents	19 days ago	
□ Downloads	10 minutes ago	
□ Dropbox	a month ago	
□ Favorites	3 months ago	
□ Google Drive	8 days ago	
□ Library	3 months ago	
□ Links	2 months ago	
□ Microsoft	4 months ago	
□ MNIST_data	2 months ago	
□ Music	2 months ago	
□ MyRdata	3 months ago	
□ OneDrive	2 days ago	
□ OpenVPN	5 months ago	
□ Pictures	24 days ago	

Data Analytics Practitioner 4: Classroom based training that allows participants to learn programming language such as Python

- Preview:
  - Using Anaconda Python Distribution (free download)



Python Language Basics, IPython, and Jupyter Notebooks

In [ ]:

```
import numpy as np
np.random.seed(12345)
np.set_printoptions(precision=4, suppress=True)
```

The Python Interpreter

```
$ python
Python 3.6.0 | packaged by conda-forge | (default, Jan 13 2017, 23:17:12)
[GCC 4.8.2 20140120 (Red Hat 4.8.2-15)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 5
>>> print(a)
5
```

```
print('Hello world')
```

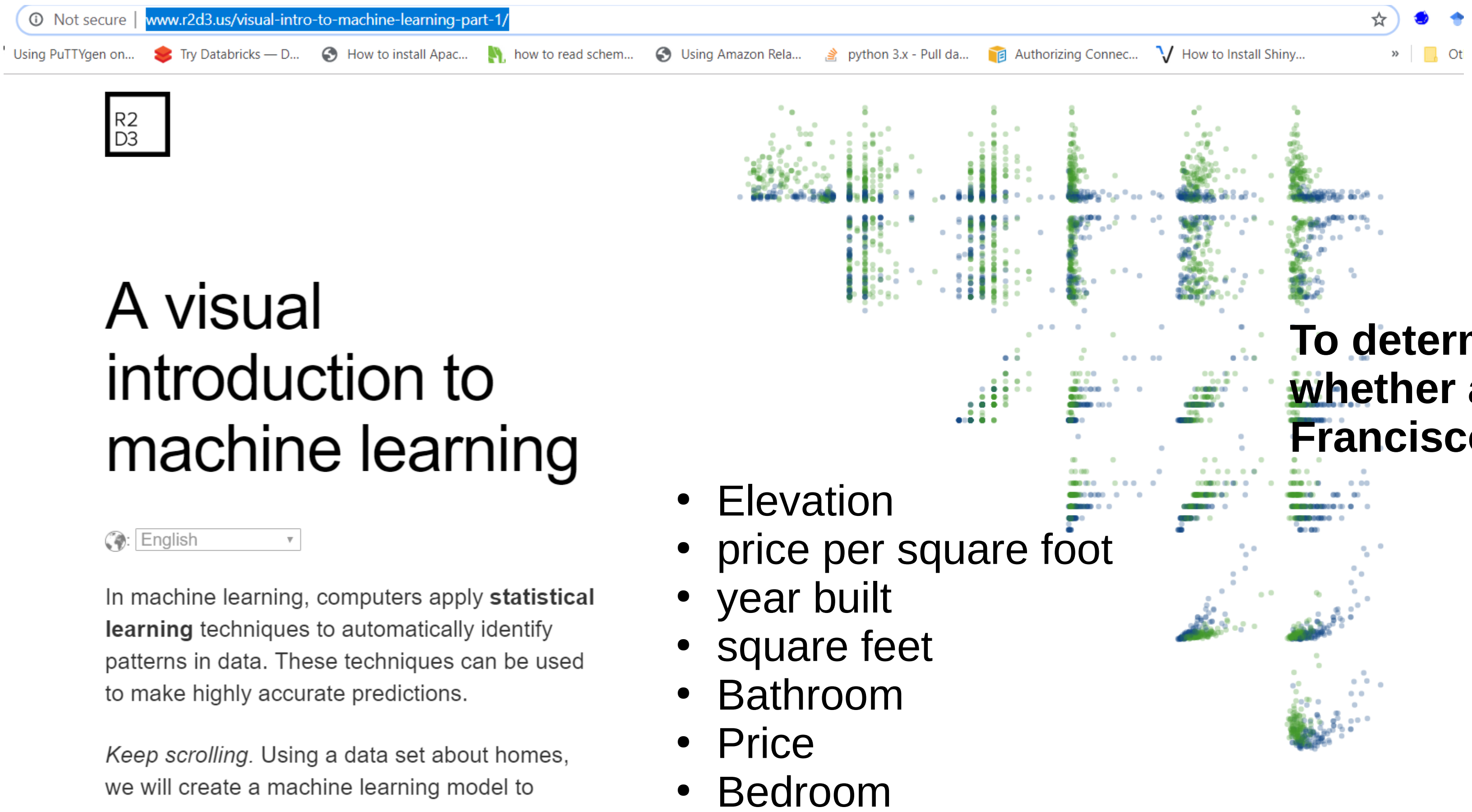
```
$ python hello_world.py
Hello world
```

Data Analytics Practitioner 4: Classroom based training that allows participants to learn programming language such as Python

Data Analytics Practitioner 5: Classroom based training that allows participants to learn programming language such as Python

- Preview:
  - Decision Tree: low cost, high accuracy and high interpretability

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/



A visual introduction to machine learning

In machine learning, computers apply **statistical learning** techniques to automatically identify patterns in data. These techniques can be used to make highly accurate predictions.

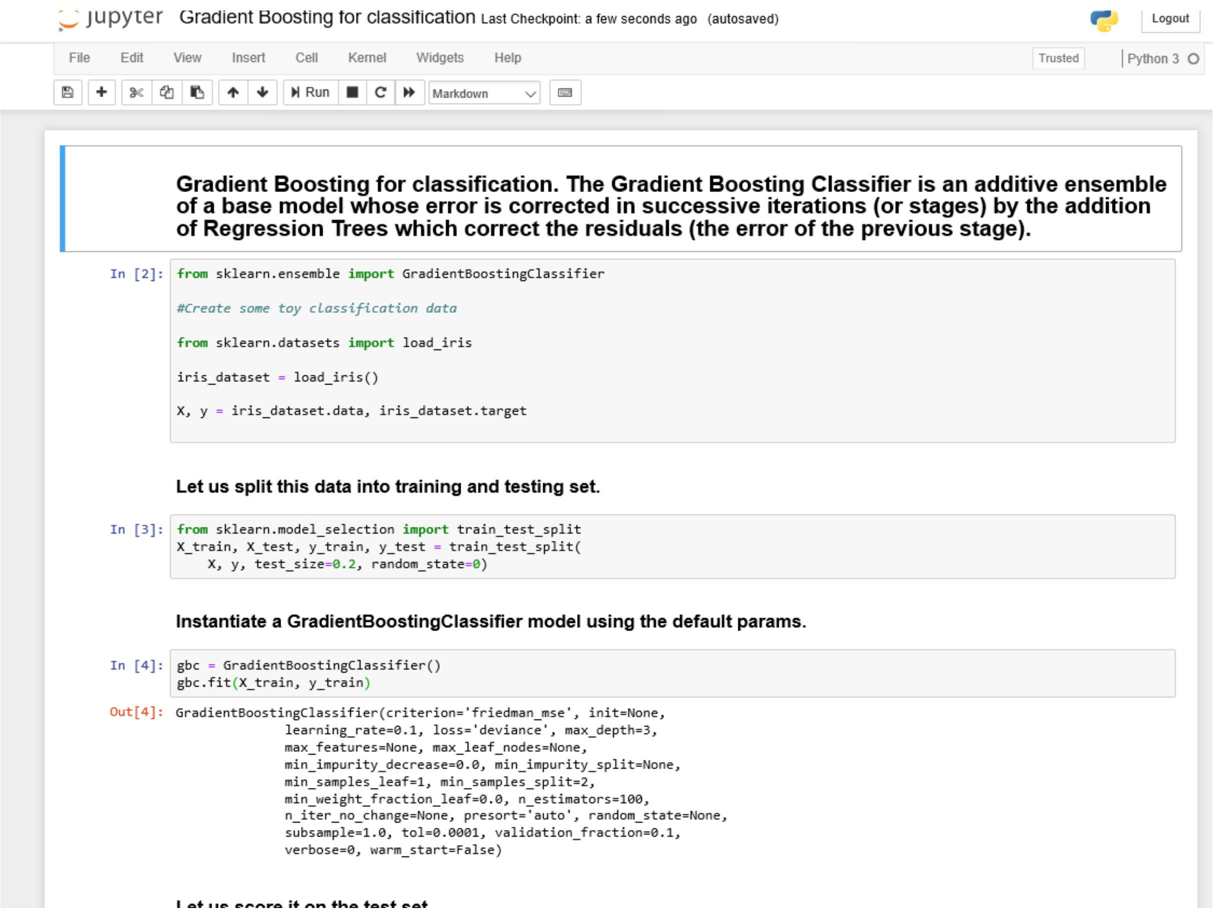
Keep scrolling. Using a data set about homes, we will create a machine learning model to

- Elevation
- price per square foot
- year built
- square feet
- Bathroom
- Price
- Bedroom

To determine/classify whether a home is in San Francisco or in New York

Data Analytics Practitioner 5: Classroom based training that allows participants to learn programming language such as Python

- Preview:
  - Gradient boosting method example – simple example on how to use a machine learning model in python



jupyter
Gradient Boosting for classification Last Checkpoint: a few seconds ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Markdown

Gradient Boosting for classification. The Gradient Boosting Classifier is an additive ensemble of a base model whose error is corrected in successive iterations (or stages) by the addition of Regression Trees which correct the residuals (the error of the previous stage).

In [2]:
```
from sklearn.ensemble import GradientBoostingClassifier

#Create some toy classification data

from sklearn.datasets import load_iris

iris_dataset = load_iris()

X, y = iris_dataset.data, iris_dataset.target
```

Let us split this data into training and testing set.

In [3]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=0)
```

Instantiate a GradientBoostingClassifier model using the default params.

In [4]:
```
gbc = GradientBoostingClassifier()
gbc.fit(X_train, y_train)
```

Out[4]:
```
GradientBoostingClassifier(criterion='friedman_mse', init=None,
              learning_rate=0.1, loss='deviance', max_depth=3,
              max_features=None, max_leaf_nodes=None,
              min_impurity_decrease=0.0, min_impurity_split=None,
              min_samples_leaf=1, min_samples_split=2,
              min_weight_fraction_leaf=0.0, n_estimators=100,
              n_iter_no_change=None, presort='auto', random_state=None,
              subsample=1.0, tol=0.0001, validation_fraction=0.1,
              verbose=0, warm_start=False)
```

Let us score it on the test set

Data Analytics Practitioner 5: Classroom based training that allows participants to learn programming language such as Python

- Preview:
  - Predictive maintenance use case

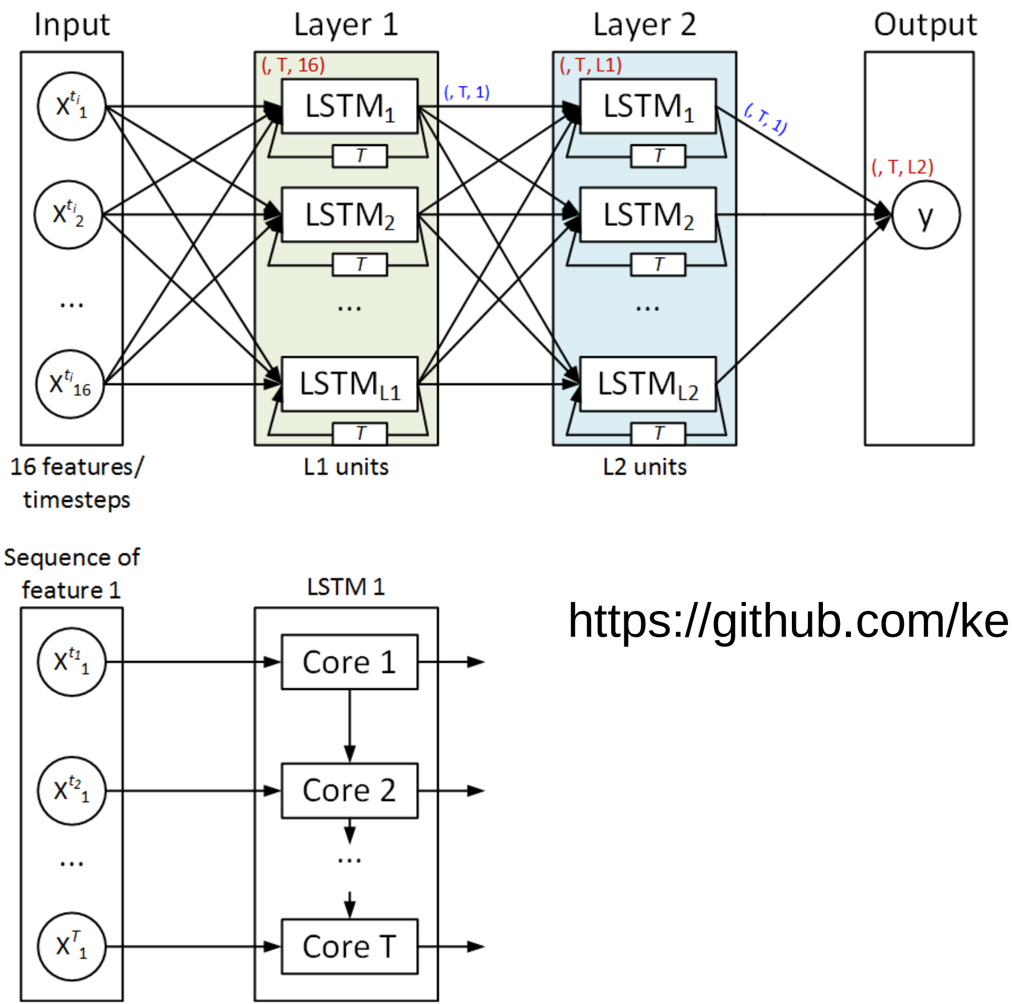
In this example I build an LSTM network in order to predict remaining useful life (or time to failure) of machines. The network uses machine sensor values to predict when an aircraft engine will fail in the future so that maintenance can be planned in advance.

The question to ask is "Given these machine operation and failure events history, can we predict when an running machine will fail?" We re-formulate this question into two closely relevant questions and answer them using two different types of machine learning models:

* Regression models: How many more cycles a running engine will last before it fails?
* Binary classification: Is this machine going to fail within w1 cycles?

Data Analytics Practitioner 5: Classroom based training that allows participants to learn programming language such as Python

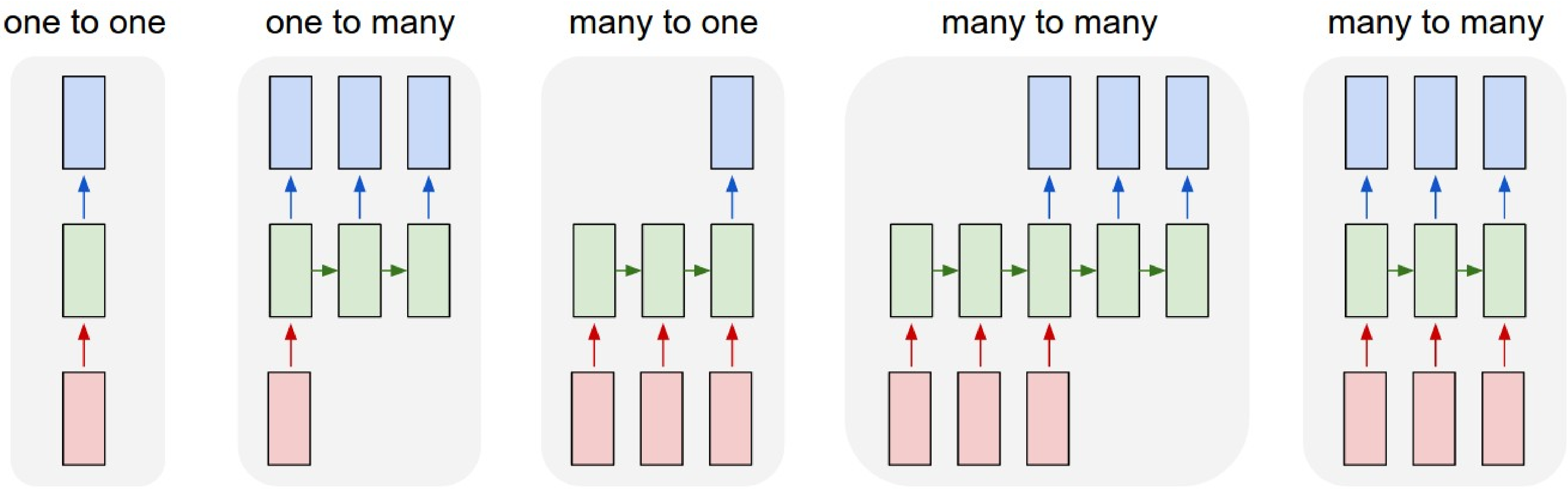
- Preview:
  - Predictive maintenance use case



https://github.com/keras-team/keras/issues/4149

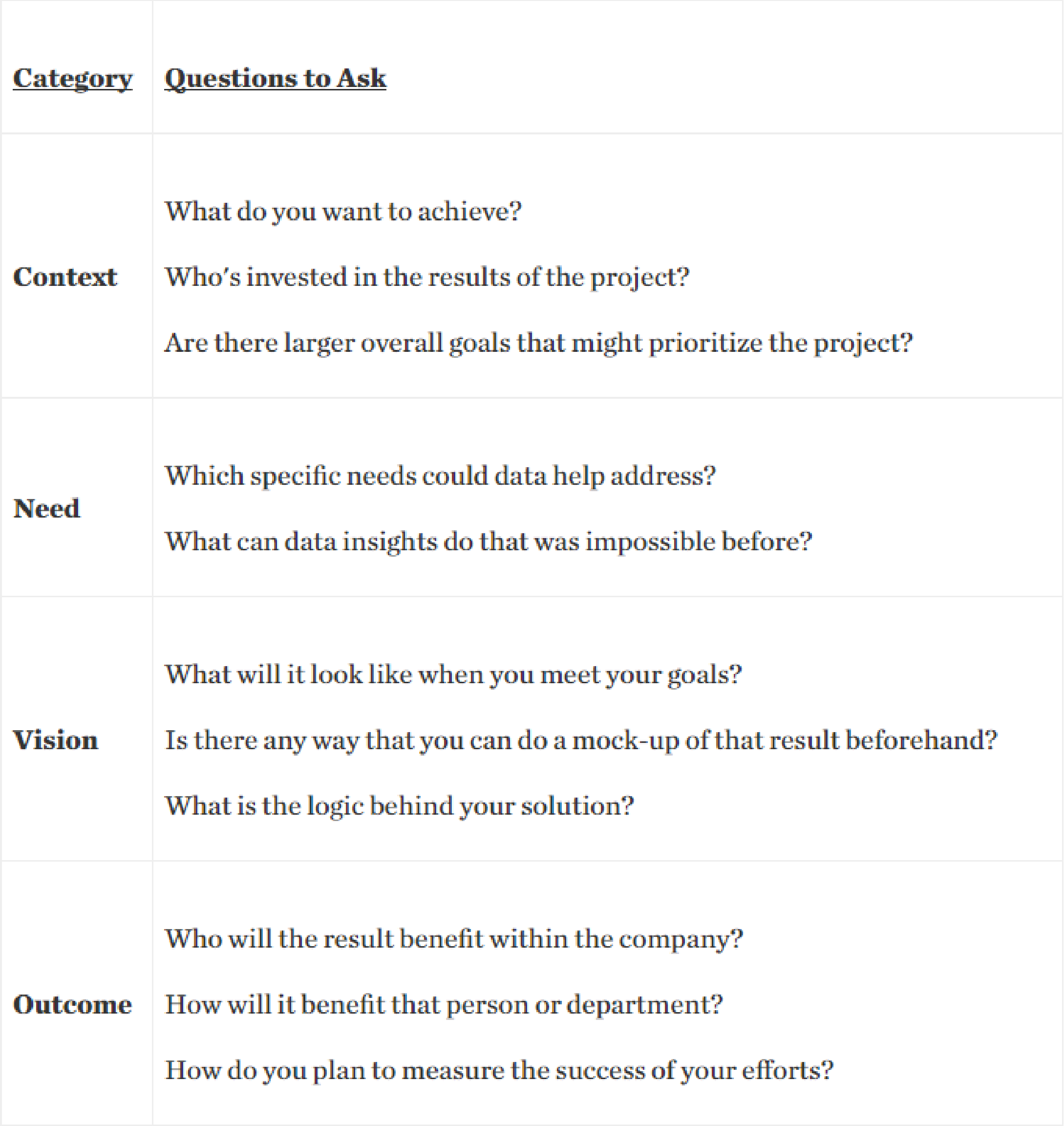
Data Analytics Practitioner 5: Classroom based training that allows participants to learn programming language such as Python

- Preview:
  - Predictive maintenance use case



Data Analytics Practitioner 6: Consultation session to help kick start client’s analytics project

- Set clear goals
  - To discuss the project scopes
  - To discuss areas out of scope, project dependencies and project success criteria
  - To identify project organization structure – key persons and roles who will be involved in the project and their project commitment



Category	Questions to Ask
Context	What do you want to achieve? Who’s invested in the results of the project? Are there larger overall goals that might prioritize the project?
Need	Which specific needs could data help address? What can data insights do that was impossible before?
Vision	What will it look like when you meet your goals? Is there any way that you can do a mock-up of that result beforehand? What is the logic behind your solution?
Outcome	Who will the result benefit within the company? How will it benefit that person or department? How do you plan to measure the success of your efforts?

Data Analytics Practitioner 6: Consultation session to help kick start client's analytics project

- Build the framework
  - To discuss the software products and technologies to be used in the project
  - To dicuss external integration and interfaces of data sources
  - To discuss the enviroment required to deliver the project
  - To discuss project approach, timeline and deliverable acceptance

Data Analytics Practitioner 6: Consultation session to help kick start client's analytics project

- Give raw data some context and bring data to life
  - Data to be used and the types of data sources
  - Correction of any data quality issues
  - Data obfuscation issue
  - Process of executing the ETL
  - Identify and implement relevant Machine learning model
  - Publication of machine learning model results to Power Bi online service