

# Python for Accounting Applications

# Python Programming and Data Science

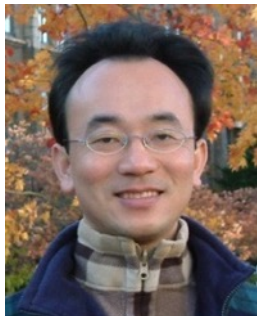
1141PAA02

ACC2, NTPU (U2004) (Fall 2025)

Wed 6, 7, 8, (14:10-17:00) (9:10-12:00) (B3F10)

 **NVIDIA**  
University Ambassador  
Certified Instructor

 **aws** educate | Cloud  
Ambassador  
2020 Cohort



Min-Yuh Day, Ph.D,  
Professor and Director

Institute of Information Management, National Taipei University

<https://web.ntpu.edu.tw/~myday>

2025-09-17



# Syllabus

**Week Date Subject/Topics**

**1 2025/09/10 Introduction to Python for Accounting Applications**

**2 2025/09/17 Python Programming and Data Science**

**3 2025/09/24 Foundations of Python Programming**

**4 2025/10/01 Data Structures**

**5 2025/10/08 Control Logic and Loops**

**6 2025/10/15 Functions and Modules; Files and Exception Handling**

**7 2025/10/22 Data Analytics and Visualization with Python**

**8 2025/10/29 Self-Learning**

# Syllabus

**Week Date Subject/Topics**

**9 2025/11/05 Midterm Project Report**

**10 2025/11/12 Obtaining Data From the Web with Python**

**11 2025/11/19 Statistical Analysis with Python**

**12 2025/11/26 Machine Learning with Python**

**13 2025/12/03 Text Analytics with Generative AI and Python**

**14 2025/12/10 Applications of Accounting Data Analytics with Python**

**15 2025/12/17 Applications of ESG Data Analytics with Python**

**16 2025/12/24 Final Project Report**

# **Python Programming and Data Science**

# Outline

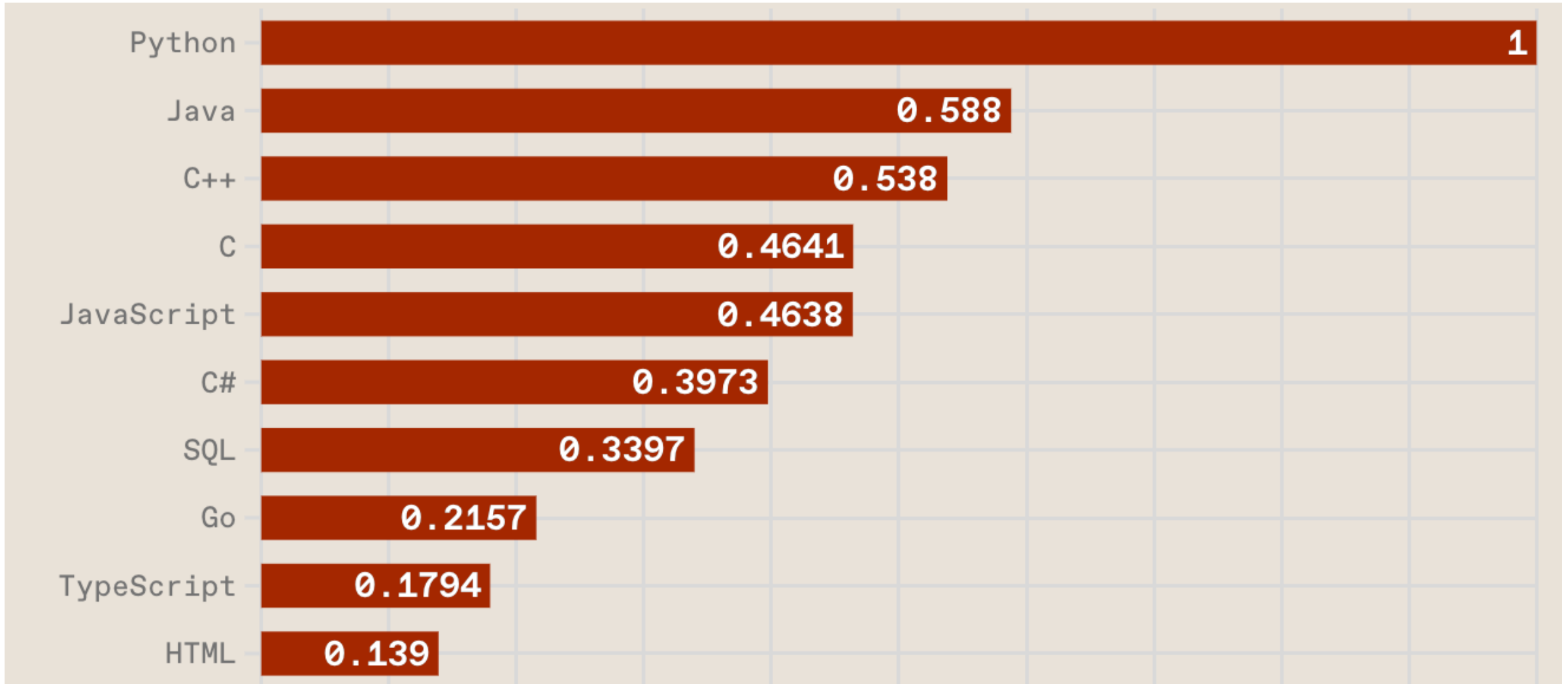
- **Python Programming**
- **Data Science**



# Python

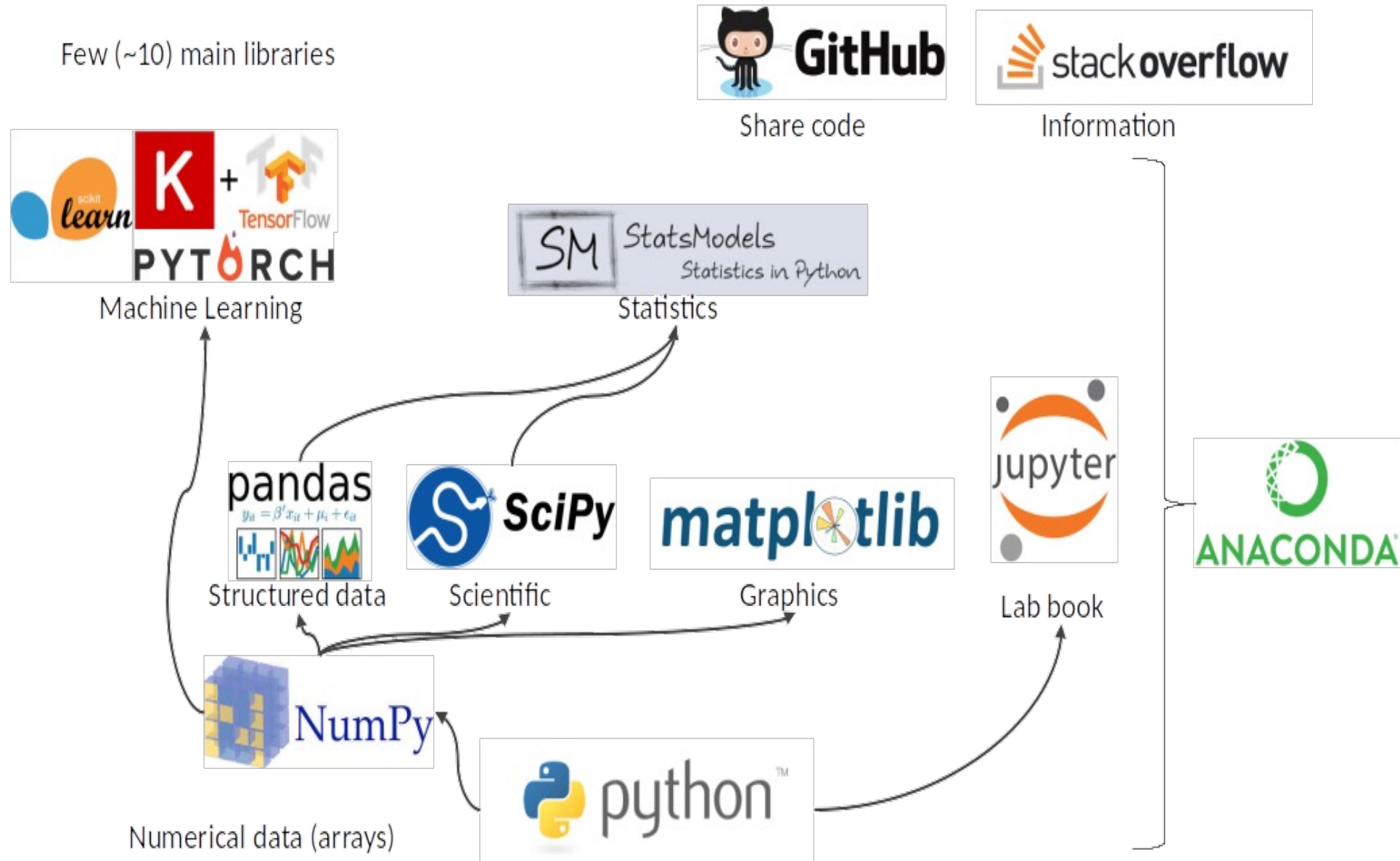
# Programming

# Top Programming Languages

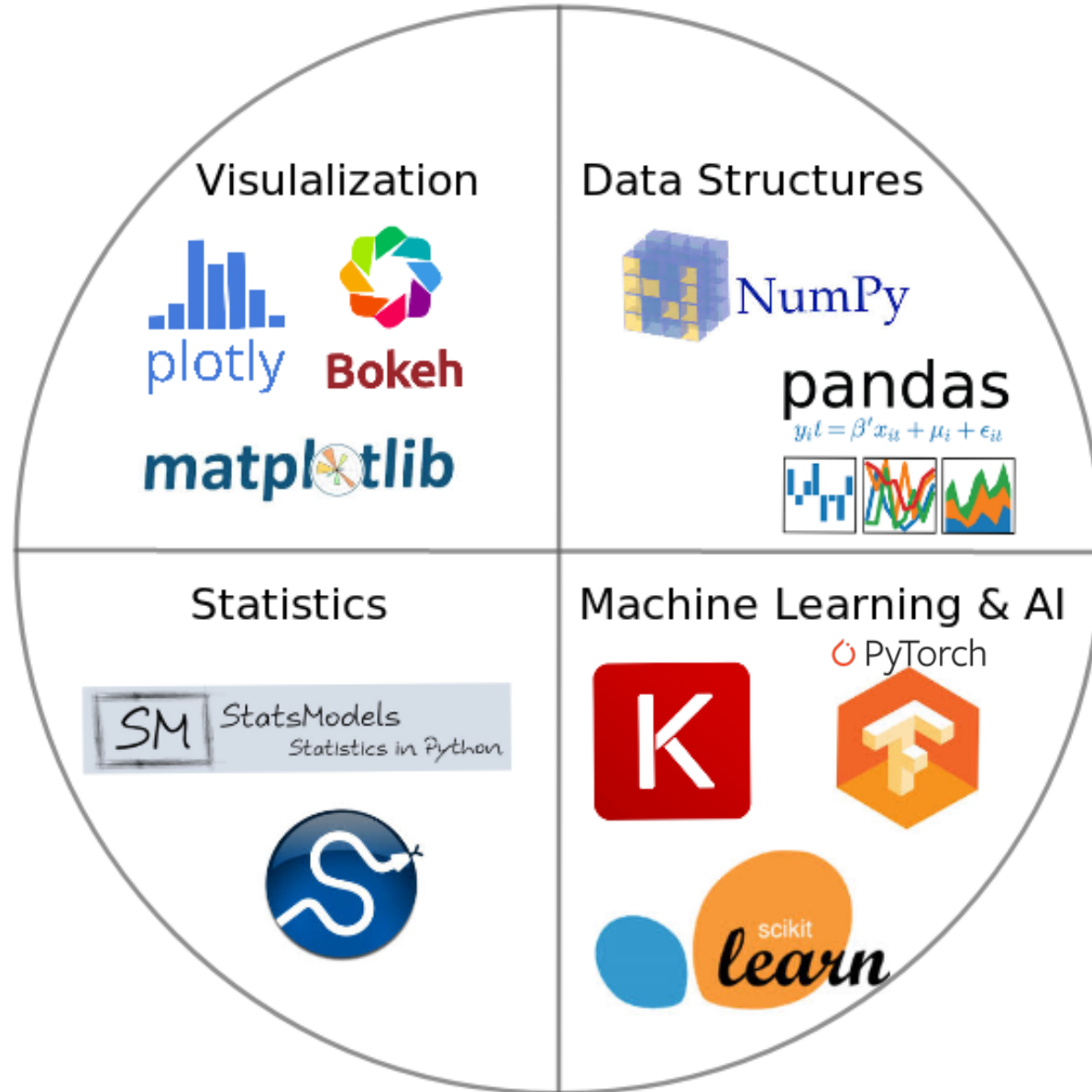


**Python** is an  
interpreted,  
object-oriented,  
high-level  
programming language  
with  
dynamic semantics.

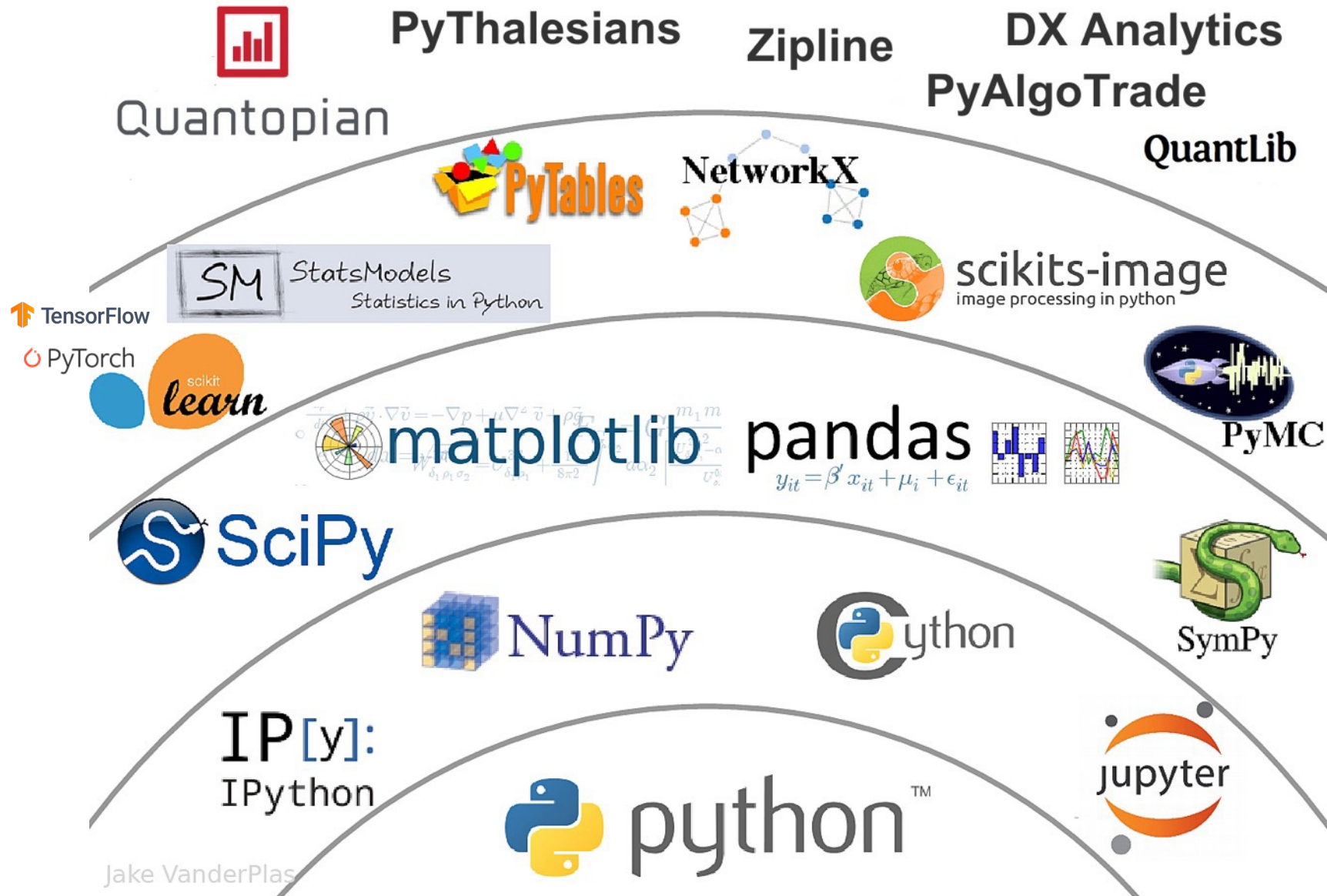
# Python Ecosystem for Data Science



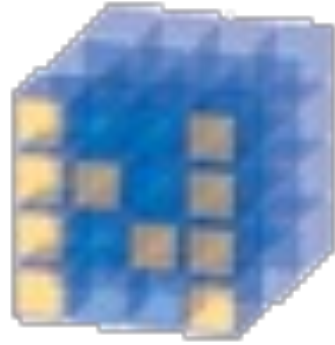
# Python Ecosystem for Data Science



# The Quant Finance PyData Stack



# NumPy



NumPy

Base

**N-dimensional array**  
package

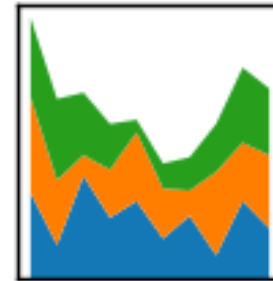
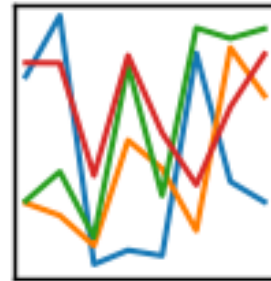
**Python**  
**matplotlib**  
**matplotlib**

# Python

# Pandas

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



- Python Tutorial
- Python HOME**
- Python Intro
- Python Get Started
- Python Syntax
- Python Comments
- Python Variables
- Python Data Types
- Python Numbers
- Python Casting
- Python Strings
- Python Booleans
- Python Operators
- Python Lists
- Python Tuples
- Python Sets
- Python Dictionaries
- Python If...Else
- Python While Loops
- Python For Loops
- Python Functions

## Python Tutorial

[← Home](#)

[Next >](#)

### Learn Python

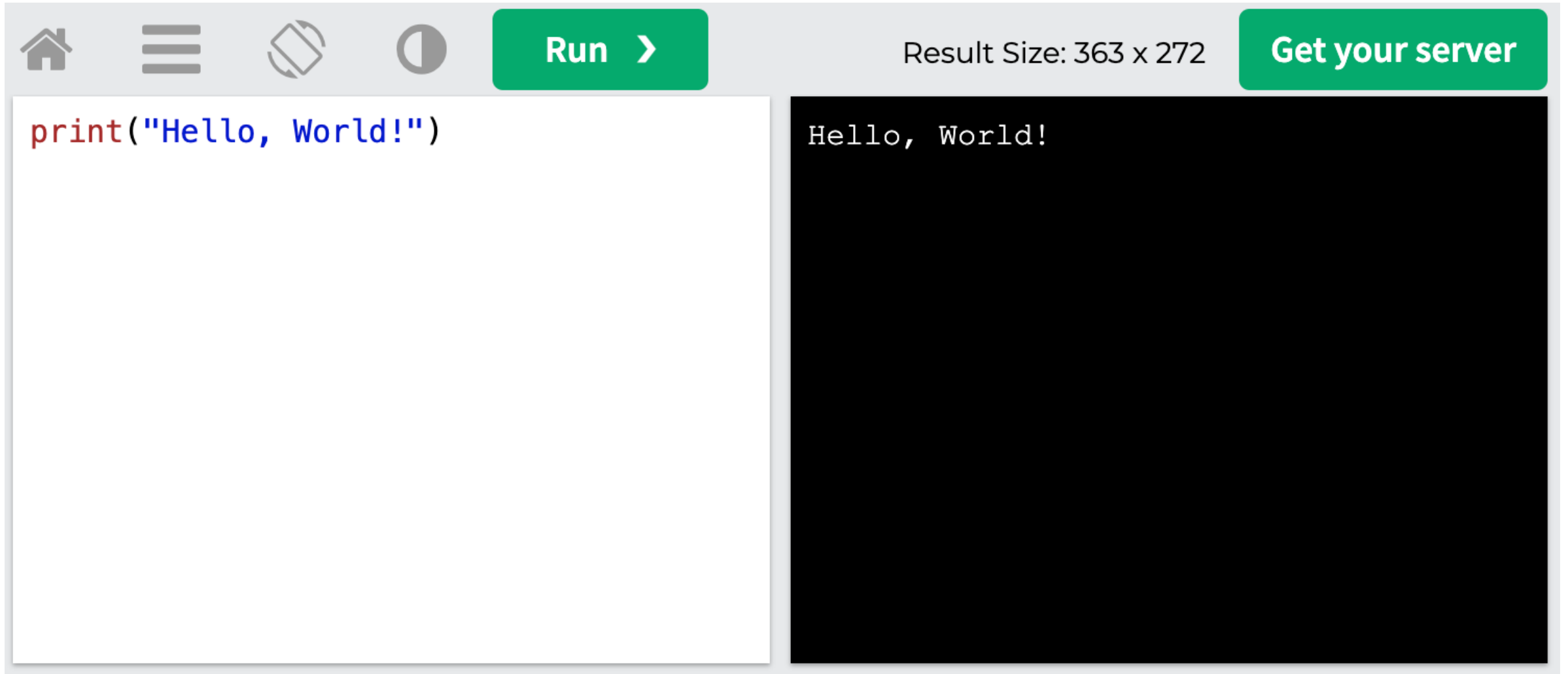
Python is a popular programming language.  
Python can be used on a server to create web applications.

[Start learning Python now »](#)

### Learning by Examples

With our "Try it Yourself" editor, you can edit Python code and view the result.

# W3Schools Python: Try Python

A screenshot of the W3Schools Python Try Python interface. The interface has a light gray header with navigation icons (home, menu, refresh, moon) and a green 'Run >' button. To the right of the 'Run' button, it says 'Result Size: 363 x 272' and a green button that says 'Get your server'. Below the header, there is a white text area on the left containing the Python code `print("Hello, World!")` and a black output area on the right displaying the result 'Hello, World!'.

# LearnPython.org



learnpython.org

[Home](#)

[About](#)

[Certify](#)

[More Languages](#) ▾

[Python](#)

[Java](#)

[HTML](#)

[Go](#)

[C](#)

[C++](#)

[JavaScript](#)

[PHP](#)

[Shell](#)

[C#](#)

[Perl](#)

[Ruby](#)

[Scala](#)

[SQL](#)

Get started learning Python with [DataCamp's](#) free [Intro to Python tutorial](#). Learn Data Science by completing interactive coding challenges and watching videos by expert instructors. [Start Now!](#)

Ready to take the test? Head onto [LearnX](#) and get your Python Certification!

This site is generously supported by [DataCamp](#). DataCamp offers online interactive [Python Tutorials](#) for Data Science. Join **11 millions** other learners and get started learning Python for data science today!

Good news! You can save 25% off your Datacamp annual subscription with the code [LEARNPYTHON23ALE25](#) - [Click here to redeem your discount!](#)

## Welcome

Welcome to the LearnPython.org interactive Python tutorial.

Whether you are an experienced programmer or not, this website is intended for everyone who wishes to learn the Python programming language.

You are welcome to join our group on [Facebook](#) for questions, discussions and updates.

After you complete the tutorials, you can get certified at [LearnX](#) and add your certification to your LinkedIn profile.

Just click on the chapter you wish to begin from, and follow the instructions. Good luck!

<https://www.learnpython.org/>

# Google's Python Class

Google for Education > Python

Search

English



Filter

## Overview

Python Set Up

Python Intro

Strings

Lists

Sorting

Dicts and Files

Regular Expressions

Utilities

## Lecture Videos

1.1 Introduction, strings [↗](#)

1.2 Lists and sorting [↗](#)

1.3 Dicts and files [↗](#)

2.1 Regular expr [↗](#)

2.2 Utilities [↗](#)

2.3 Utilities urllib [↗](#)

2.4 Conclusions [↗](#)

## Python Exercises



Home > Products > Google for Education > Python

Was this helpful? [👍](#) [🗨️](#)

## Google's Python Class [📄](#)

Welcome to Google's Python Class -- this is a free class for people with a little bit of programming experience who want to learn Python. The class includes written materials, lecture videos, and lots of code exercises to practice Python coding. These materials are used within Google to introduce Python to people who have just a little programming experience. The first exercises work on basic Python concepts like strings and lists, building up to the later exercises which are full programs dealing with text files, processes, and http connections. The class is geared for people who have a little bit of programming experience in some language, enough to know what a "variable" or "if statement" is. Beyond that, you do not need to be an expert programmer to use this material.

To get started, the Python sections are linked at the left -- [Python Set Up](#) to get Python installed on your machine, [Python Introduction](#) for an introduction to the language, and then [Python Strings](#) starts the coding material, leading to the first exercise. The end of each written section includes a link to the code exercise for that section's material. The lecture videos parallel the written materials, introducing Python, then strings, then first exercises, and so on. At Google, all this material makes up an intensive 2-day class, so the videos are organized as the day-1 and day-2 sections.

This material was created by [Nick Parlante](#) working in the engEDU group at Google. Special thanks for the help from my Google colleagues John Cox, Steve Glassman, Piotr Kaminski, and Antoine Picard. And finally thanks to Google and my director Maggie Johnson for the enlightened generosity to put these materials out on the internet for free under the [Creative Commons Attribution 2.5](#) license -- share and enjoy!

<https://developers.google.com/edu/python>

# Google Colab

Table of contents

- Getting Started
- Highlighted Features
  - TensorFlow execution
- GitHub
- Visualization
- Forms
- Examples
- Local runtime support

SECTION

## Welcome to Colaboratory!

Colaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud. See our [FAQ](#) for more info.

### Getting Started

- [Overview of Colaboratory](#)
- [Loading and saving data: Local files, Drive, Sheets, Google Cloud Storage](#)
- [Importing libraries and installing dependencies](#)
- [Using Google Cloud BigQuery](#)
- [Forms, Charts, Markdown, & Widgets](#)
- [TensorFlow with GPU](#)
- [Machine Learning Crash Course: Intro to Pandas & First Steps with TensorFlow](#)

### Highlighted Features

#### Seedbank

Looking for Colab notebooks to learn from? Check out [Seedbank](#), a place to discover interactive machine learning examples.

### TensorFlow execution

Colaboratory allows you to execute TensorFlow code in your browser with a single click. The example below adds two matrices.

$$\begin{bmatrix} 1. & 1. & 1. \end{bmatrix} + \begin{bmatrix} 1. & 2. & 3. \end{bmatrix} = \begin{bmatrix} 2. & 3. & 4. \end{bmatrix}$$

# Connect Google Colab in Google Drive

The image shows a browser window with the Google Drive interface. The address bar displays 'https://drive.google.com/drive/u/2/my-drive'. The main navigation bar includes the Drive logo, a search bar, and utility icons. On the left sidebar, the 'New' button is highlighted with a red dashed box. A dropdown menu is open, listing options like 'New folder...', 'Upload files...', 'Google Docs', 'Google Sheets', 'Google Slides', and 'More'. The 'More' option is also highlighted with a red dashed box. A second dropdown menu is open from 'More', listing 'Google Forms', 'Google Drawings', 'Google My Maps', 'Google Sites', and 'Connect more apps'. The 'Connect more apps' option is highlighted with a red dashed box. The main content area shows a 'Files' section with a 'Name' column header and an upward arrow. A storage usage indicator shows '0 bytes of 15 GB used' with an 'UPGRADE STORAGE' link. A notification for 'Get Backup and Sync for Mac' is visible at the bottom left.

# Google Colab

The screenshot shows the Google Drive interface with a 'Connect apps to Drive' dialog box open. The dialog box has a search bar at the top with 'colab' entered and highlighted by a red dashed border. Below the search bar, there are six app cards arranged in a 2x3 grid:

- ZIP Extractor**: Extract ZIP files to Google Drive. Extraction complete. 307,585 users.
- Lumin PDF**: Beautiful PDF Editor. 289,310 users.
- CloudConvert**: 373,161 users.
- Sejda**: Merge PDF - Split PDF - Sejda.com. 1106 reviews.
- DocHub**: Edit and Sign PDF Documents. 2,131,600 users.
- Google Forms**: 4,803,614 users.

The background shows the Google Drive sidebar with options like 'My Drive', 'Computers', 'Shared with me', 'Recent', 'Starred', 'Trash', 'Backups', and 'Storage'. The top navigation bar includes a search bar and various utility icons.

# Google Colab

The image shows a browser window with a Google Drive page. A modal dialog titled "Connect apps to Drive" is open in the center. The dialog has a search bar containing "colab". Below the search bar, a list of apps is shown. The first app, "Colaboratory", is highlighted with a red dashed border. The app's details include a logo of two yellow circles, the name "Colaboratory", the URL "https://colab.research.google.com", a description: "A data analysis tool that combines code, output, and descriptive text into one collaborative document.", and a rating of five stars with 195 reviews. A blue button with a plus sign and the text "+ CONNECT" is also highlighted with a red dashed border. The background shows the Google Drive interface with a sidebar on the left containing navigation options like "My Drive", "Computers", "Shared with me", "Recent", "Starred", "Trash", "Backups", and "Storage". The top of the browser shows the address bar with the URL "https://drive.google.com/drive/u/2/my-drive".

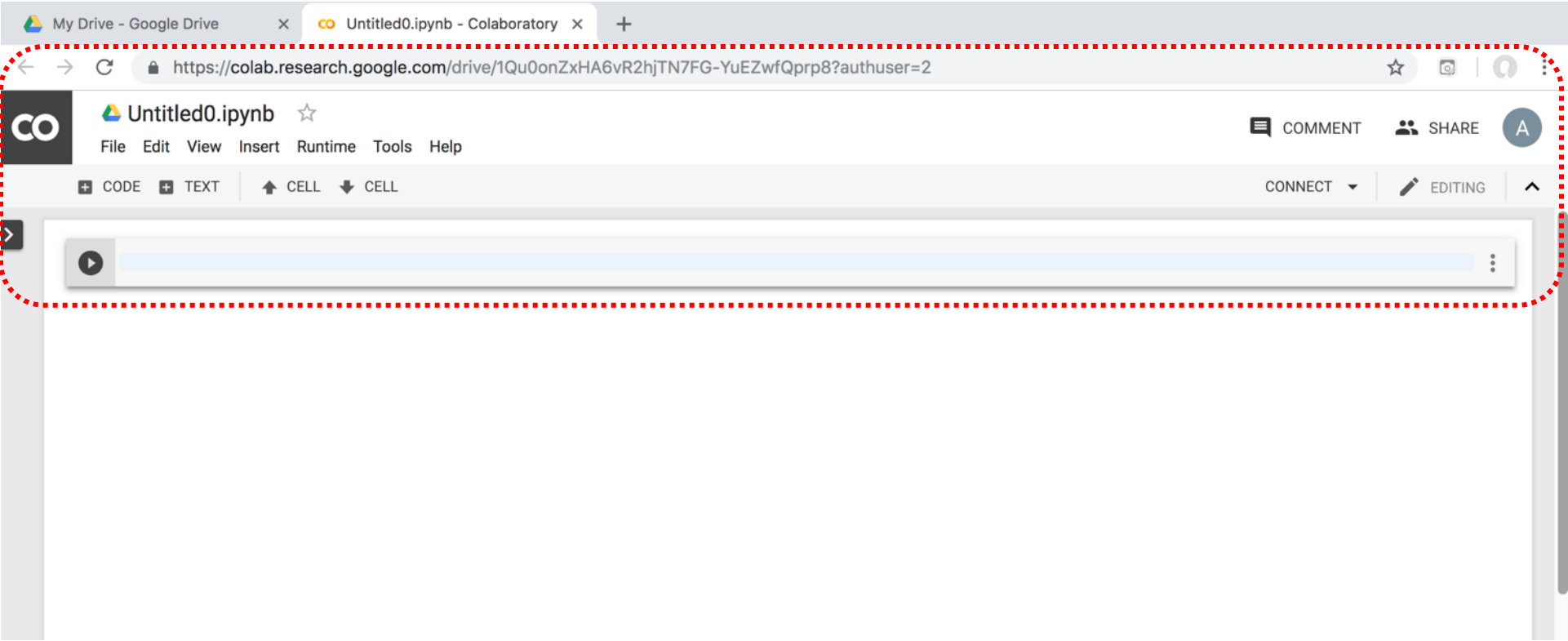
# Connect Colaboratory to Google Drive

The screenshot shows the Google Drive web interface. A dialog box titled "Connect apps to Drive" is open, displaying a search for "colab". A confirmation message from Colaboratory is centered in the dialog, stating "Colaboratory was connected to Google Drive." and "Make Colaboratory the default app for files it can open" with a checked checkbox. An "OK" button is visible at the bottom right of the message. The background shows the Drive sidebar with categories like "My Drive", "Computers", "Shared with me", "Recent", "Starred", "Trash", "Backups", and "Storage". The storage status indicates "0 bytes of 15 GB used" with an "UPGRADE STORAGE" link. The top navigation bar includes the Drive logo, search bar, and various utility icons.

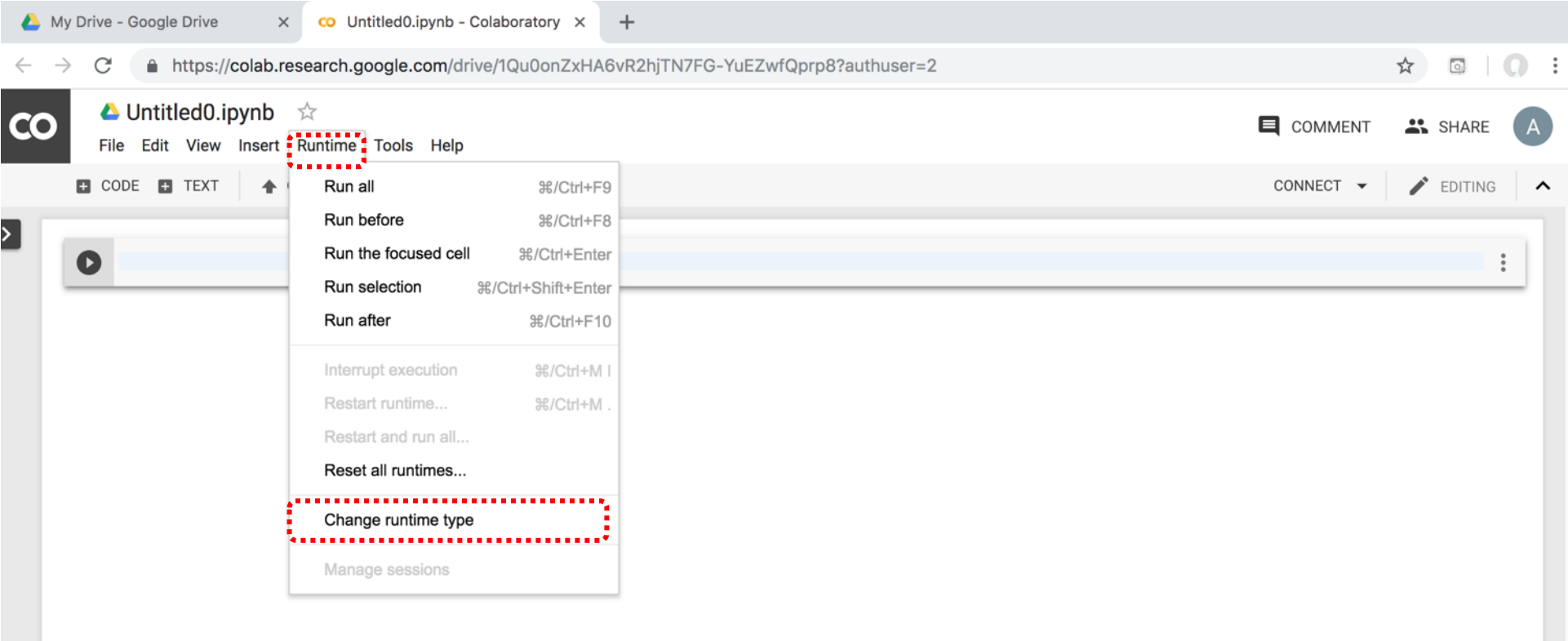
# Google Colab

The image shows a browser window with the Google Drive interface. The address bar displays the URL `https://drive.google.com/drive/u/2/my-drive`. The main header includes the Drive logo, a search bar, and navigation icons. On the left sidebar, the 'New' button is highlighted with a red dashed box. A dropdown menu is open, listing various options: 'New folder...', 'Upload files...', 'Upload folder...', 'Google Docs', 'Google Sheets', 'Google Slides', 'More', 'Google Forms', 'Google Drawings', 'Google My Maps', 'Google Sites', 'Colaboratory', and 'Connect more apps'. The 'More' option in the first dropdown and the 'Colaboratory' option in the second dropdown are both highlighted with red dashed boxes. The background shows the 'Quick Access' section with 'My Drive' selected, and a 'Files' section with storage information (0 bytes of 15 GB used) and a 'Get Backup and Sync for Mac' notification.

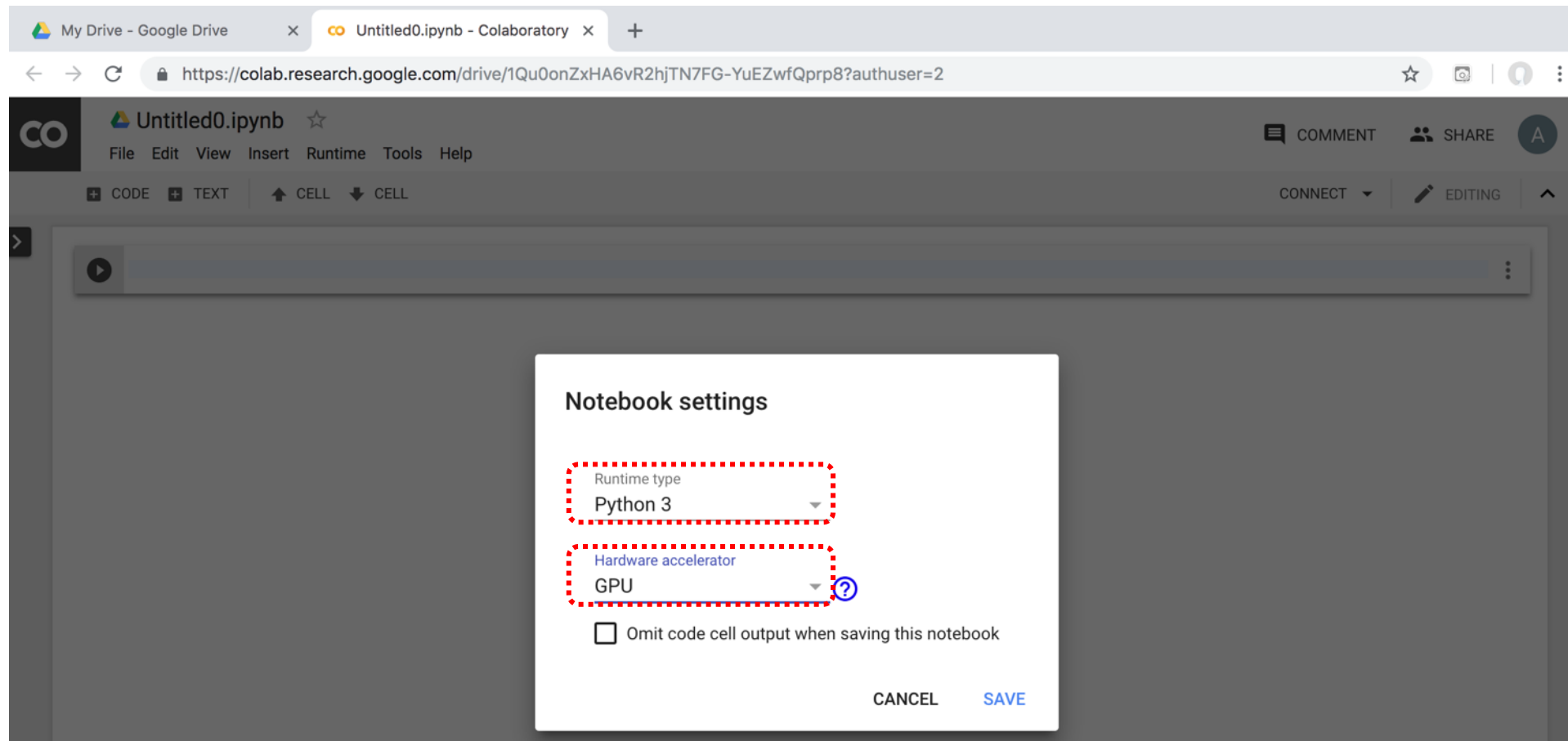
# Google Colab



# Google Colab



# Run Jupyter Notebook Python3 GPU Google Colab



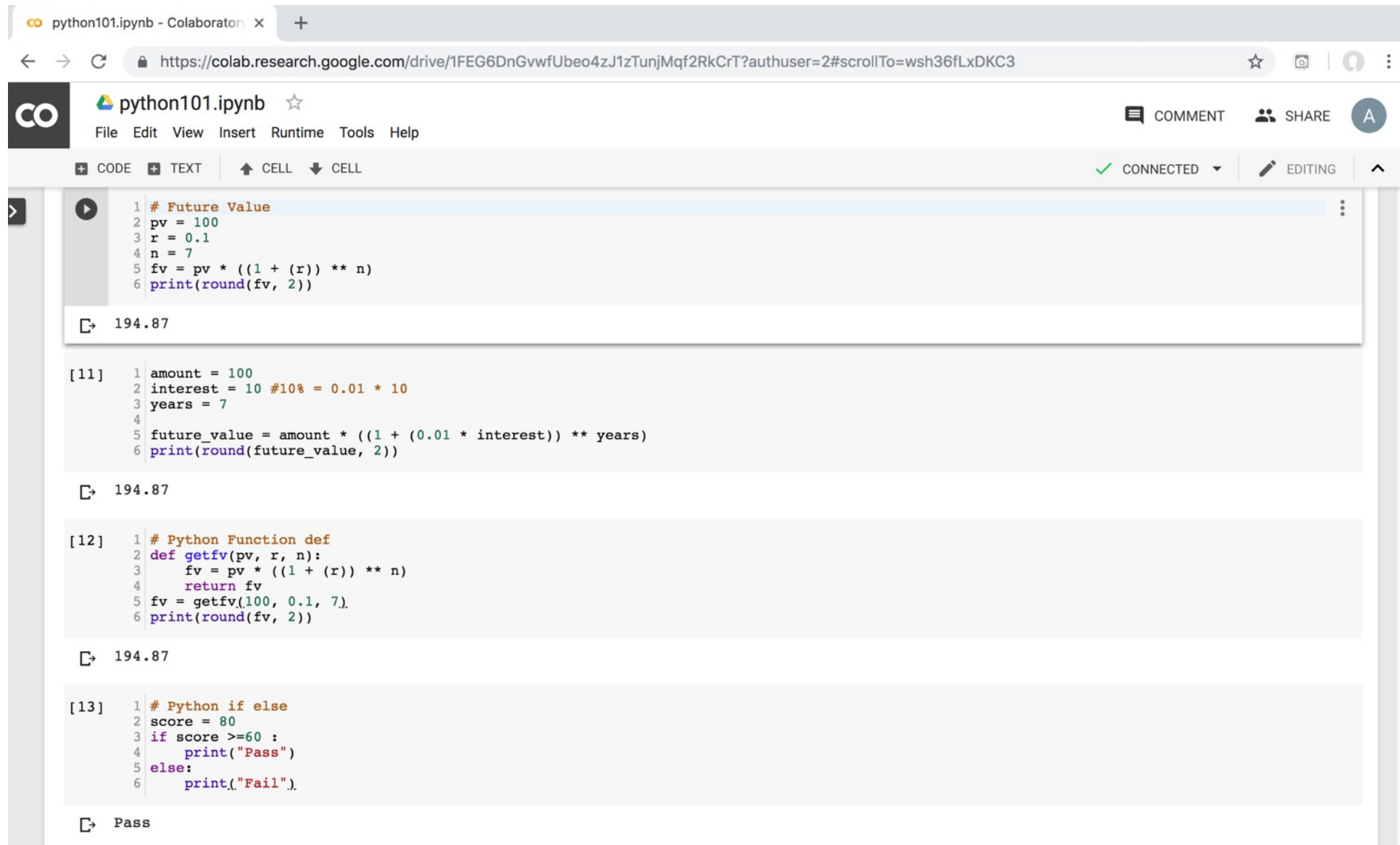
# Google Colab Python Hello World

```
print('Hello World')
```



# Python in Google Colab (Python101)

<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>



The screenshot shows a Google Colab notebook titled "python101.ipynb". The interface includes a top navigation bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help" menus. On the right, there are "COMMENT", "SHARE", and a user profile icon. Below the navigation bar, the notebook is in "EDITING" mode, and the status is "CONNECTED".

The notebook contains four code cells:

- Cell 1:** A code cell with the following Python code:

```
1 # Future Value
2 pv = 100
3 r = 0.1
4 n = 7
5 fv = pv * ((1 + (r)) ** n)
6 print(round(fv, 2))
```

The output is "194.87".
- Cell [11]:** A code cell with the following Python code:

```
1 amount = 100
2 interest = 10 #10% = 0.01 * 10
3 years = 7
4
5 future_value = amount * ((1 + (0.01 * interest)) ** years)
6 print(round(future_value, 2))
```

The output is "194.87".
- Cell [12]:** A code cell with the following Python code:

```
1 # Python Function def
2 def getfv(pv, r, n):
3     fv = pv * ((1 + (r)) ** n)
4     return fv
5 fv = getfv(100, 0.1, 7)
6 print(round(fv, 2))
```

The output is "194.87".
- Cell [13]:** A code cell with the following Python code:

```
1 # Python if else
2 score = 80
3 if score >=60 :
4     print("Pass")
5 else:
6     print("Fail").
```

The output is "Pass".

<https://tinyurl.com/aintpupython101>



**Anaconda**  
**The Most Popular**  
**Python**  
**Data Science Platform**

# Download Anaconda



Products ▾

Pricing

Solutions ▾

Resources ▾

Partners ▾

Blog

Company ▾

Contact Sales

## Data science technology for a better world.

Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine. Start working with thousands of open-source packages and libraries today.

Download 

Get Additional Installers



<https://www.anaconda.com/download>

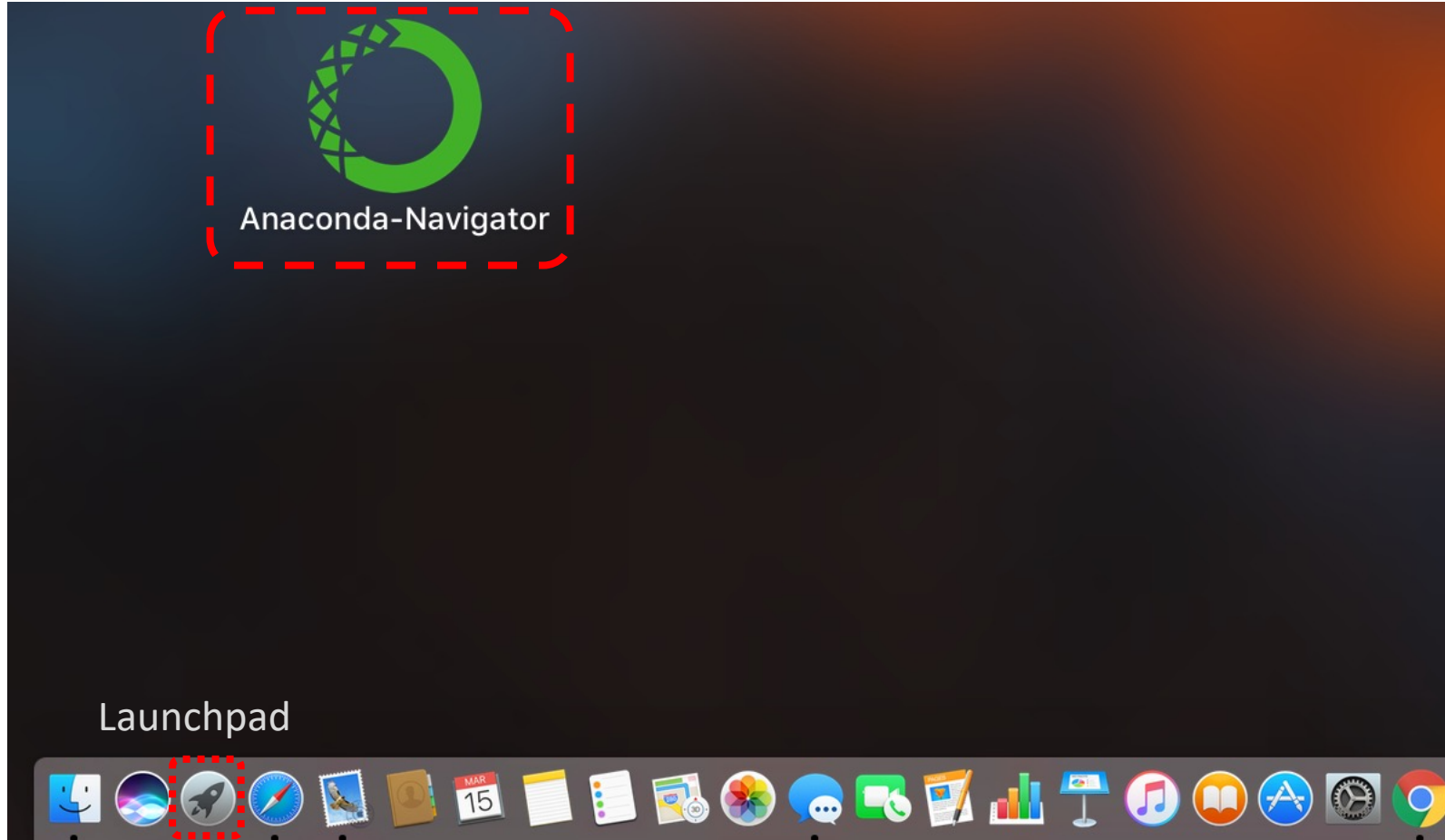




# Python

# HelloWorld

# Anaconda-Navigator



# Anaconda Navigator

The screenshot displays the Anaconda Navigator desktop application. At the top, the title bar reads "Anaconda Navigator". Below it, the application header features the "ANACONDA NAVIGATOR" logo on the left and a "Sign in to Anaconda Cloud" button on the right. A left-hand sidebar contains navigation options: "Home", "Environments", "Learning", and "Community". At the bottom of the sidebar are links for "Documentation", "Developer Blog", and "Feedback", along with social media icons for Twitter, YouTube, and GitHub.

The main content area is titled "Applications on" and shows a dropdown menu set to "base (root)" and a "Channels" button. A "Refresh" button is located in the top right of this section. The applications are arranged in a grid:

- jupyterlab** (0.31.5): An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. [Launch]
- jupyter notebook** (5.4.0): Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. [Launch]
- qtconsole** (4.3.1): PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more. [Launch]
- spyder** (3.2.6): Scientific PYTHON Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features. [Launch]
- vscode** (1.22.2): Streamlined code editor with support for development operations like debugging, task running and version control. [Launch]
- glueviz** (0.12.4): Multidimensional data visualization across files. Explore relationships within and among related datasets. [Install]

The "jupyter notebook" application card is highlighted with a red dashed border, and its "Launch" button is enclosed in a solid red box.

# Jupyter Notebook

The screenshot shows a web browser window with the Jupyter Notebook interface. The browser's address bar displays the URL `localhost:8888/tree/Documents/Data/BDA`. The Jupyter logo and a "Logout" button are visible at the top. Below the navigation tabs ("Files", "Running", "Clusters"), there is a prompt "Select items to perform actions on them." followed by "Upload", "New", and "Refresh" buttons. A file browser table is shown with a red dashed border around its content area. The table has a header with "Name" and "Last Modified" columns. The first row shows a folder icon, a selection checkbox, a dropdown menu with "0", and the path `/ Documents / Data / BDA`. The second row shows a folder icon, a double-dot `..` entry, and the text "seconds ago". A message at the bottom of the table states "The notebook list is empty."

Home x

localhost:8888/tree/Documents/Data/BDA

jupyter Logout

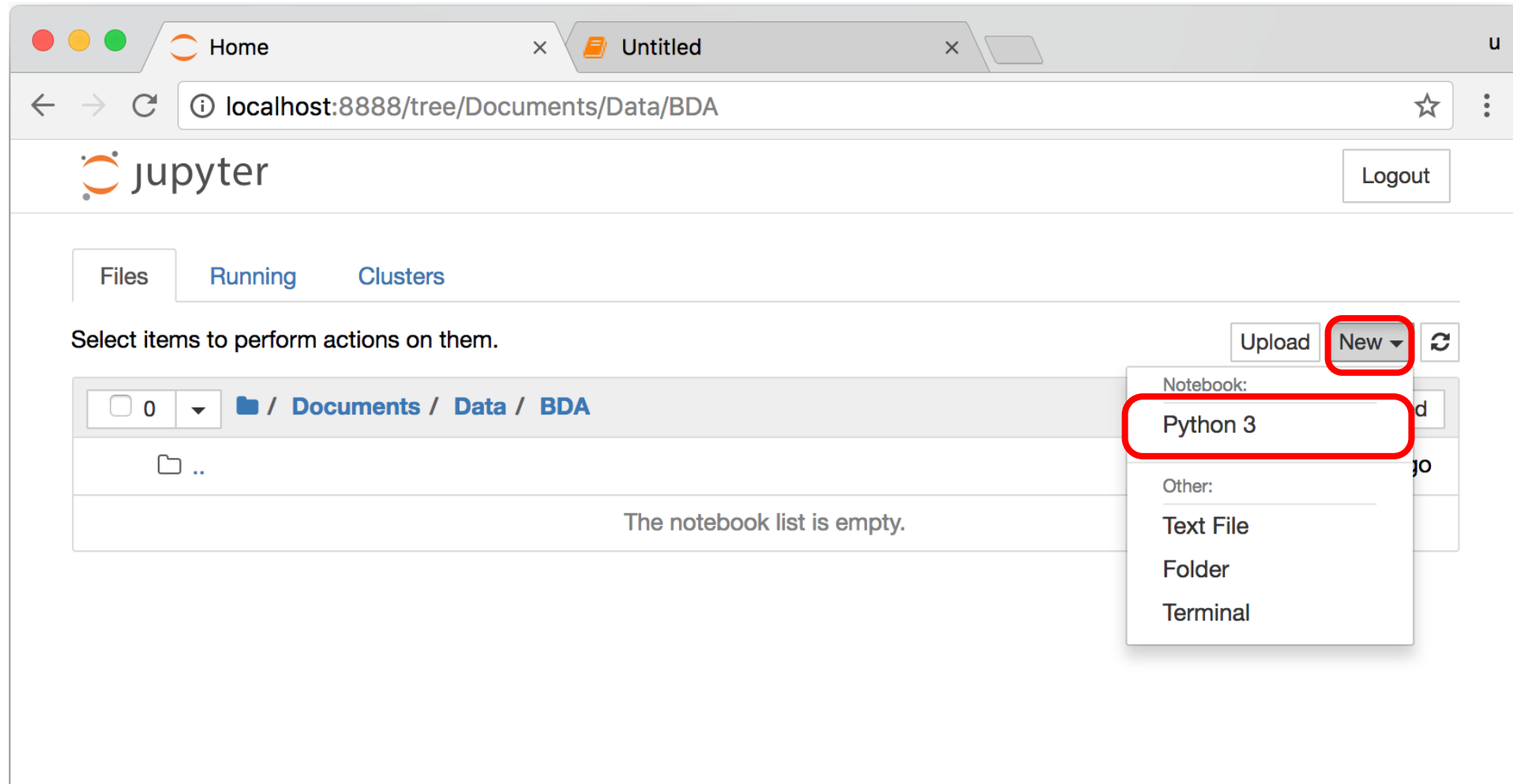
Files Running Clusters

Select items to perform actions on them. Upload New Refresh

	Name ↓	Last Modified
<input type="checkbox"/> 0	/ Documents / Data / BDA	
<input type="checkbox"/>	..	seconds ago
The notebook list is empty.		

# Jupyter Notebook

## New Python 3



The screenshot shows a web browser window with the Jupyter Notebook interface. The browser's address bar displays `localhost:8888/tree/Documents/Data/BDA`. The Jupyter logo and a "Logout" button are visible at the top. Below the logo, there are tabs for "Files", "Running", and "Clusters". A message says "Select items to perform actions on them." To the right of this message are buttons for "Upload", "New", and a refresh icon. The "New" button is highlighted with a red circle. A dropdown menu is open, showing options under "Notebook:" and "Other:". The "Python 3" option under "Notebook:" is highlighted with a red rectangle. Other options include "Text File", "Folder", and "Terminal". The file browser shows the path `/ Documents / Data / BDA` and a message that says "The notebook list is empty."

```
print("hello, world")
```

The screenshot shows a web browser window displaying a Jupyter Notebook. The browser's address bar shows the URL `localhost:8888/notebooks/Documents/Data/BDA/HelloWorld.ipynb`. The notebook's title bar reads "jupyter HelloWorld (autosaved)". The main menu includes "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". The "Run" button in the toolbar is highlighted with a red circle. Below the toolbar, a code cell is shown with the input `In [1]: print("hello, world")` and the output `hello, world`. The code and output are both highlighted with red circles. Below the code cell is an empty input field labeled `In [ ]:`.



# Python Fiddle

The screenshot shows the Python Fiddle web application in a browser window. The browser's address bar displays "pythonfiddle.com". The application's header includes navigation buttons for "Run", "Reset", "Share", "Import", and "Login", along with a "Language" dropdown menu. The Python Fiddle logo, featuring a cloud with Python icons, is positioned in the top right corner, with the text "Python Fiddle" and "Python Cloud IDE" below it. A "G+1" button and a "2.6k" counter are also visible. The main workspace is a code editor with two lines of Python code: 

```
1 print("Hello Python Fiddle")  
2
```

. On the left side, there is a sidebar with a list of "Examples" including "Chaining comparison operators", "Decorators", "Creating generators objects", "Enumerate", "Function closure", "Lex tokenizer", "Step argument in slice operators", "For Else", "Verbose regular expressions", "In-place value swapping", and "Function argument unpacking". Below the examples are buttons for "Packages" and "Hotkeys". On the right side, there is a form for saving the code, with fields for "Title:", "Description:", and "Tags:". The "Tags" field includes a note: "A comma-separated list of tags." and a "Save" button at the bottom.

Hello Python Fiddle

# Text input and output

```
print("Hello World")
```

```
print("Hello World\nThis is a message")
```

```
x = 3  
print(x)
```

```
x = 2  
y = 3  
print(x, ' ', y)
```

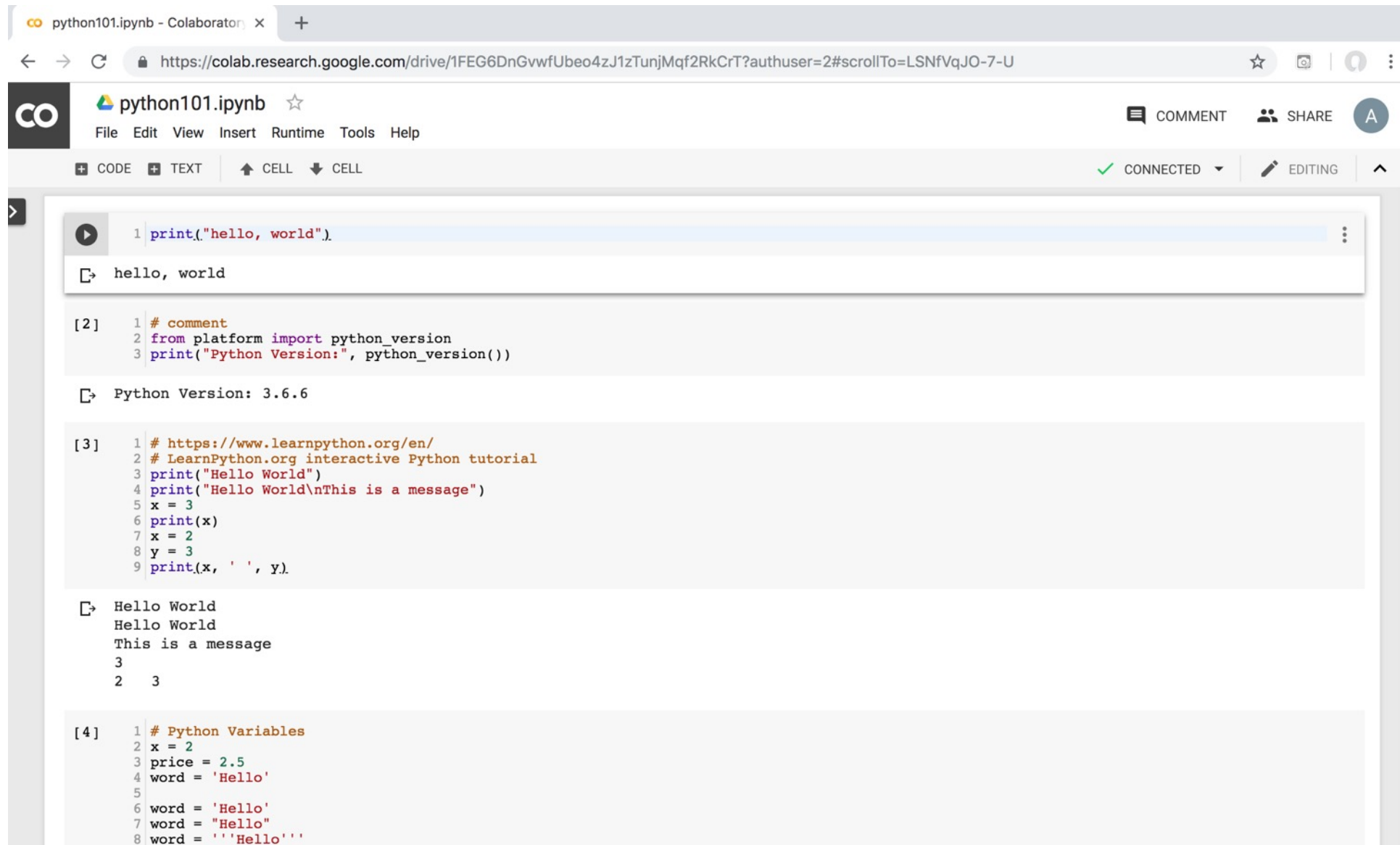
```
name = input("Enter a name: ")
```

```
x = int(input("What is x? "))
```

```
x = float(input("Write a number "))
```

# Python in Google Colab

<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>



The screenshot shows a Google Colab notebook interface. The browser address bar displays the URL: <https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT?authuser=2#scrollTo=LSNfVqJO-7-U>. The notebook title is "python101.ipynb". The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with options for CODE, TEXT, CELL, and a status indicator showing "CONNECTED" and "EDITING".

The notebook contains four code cells:

- Cell 1:** `print("hello, world").` Output: `hello, world`
- Cell 2:**

```
1 # comment
2 from platform import python_version
3 print("Python Version:", python_version())
```

 Output: `Python Version: 3.6.6`
- Cell 3:**

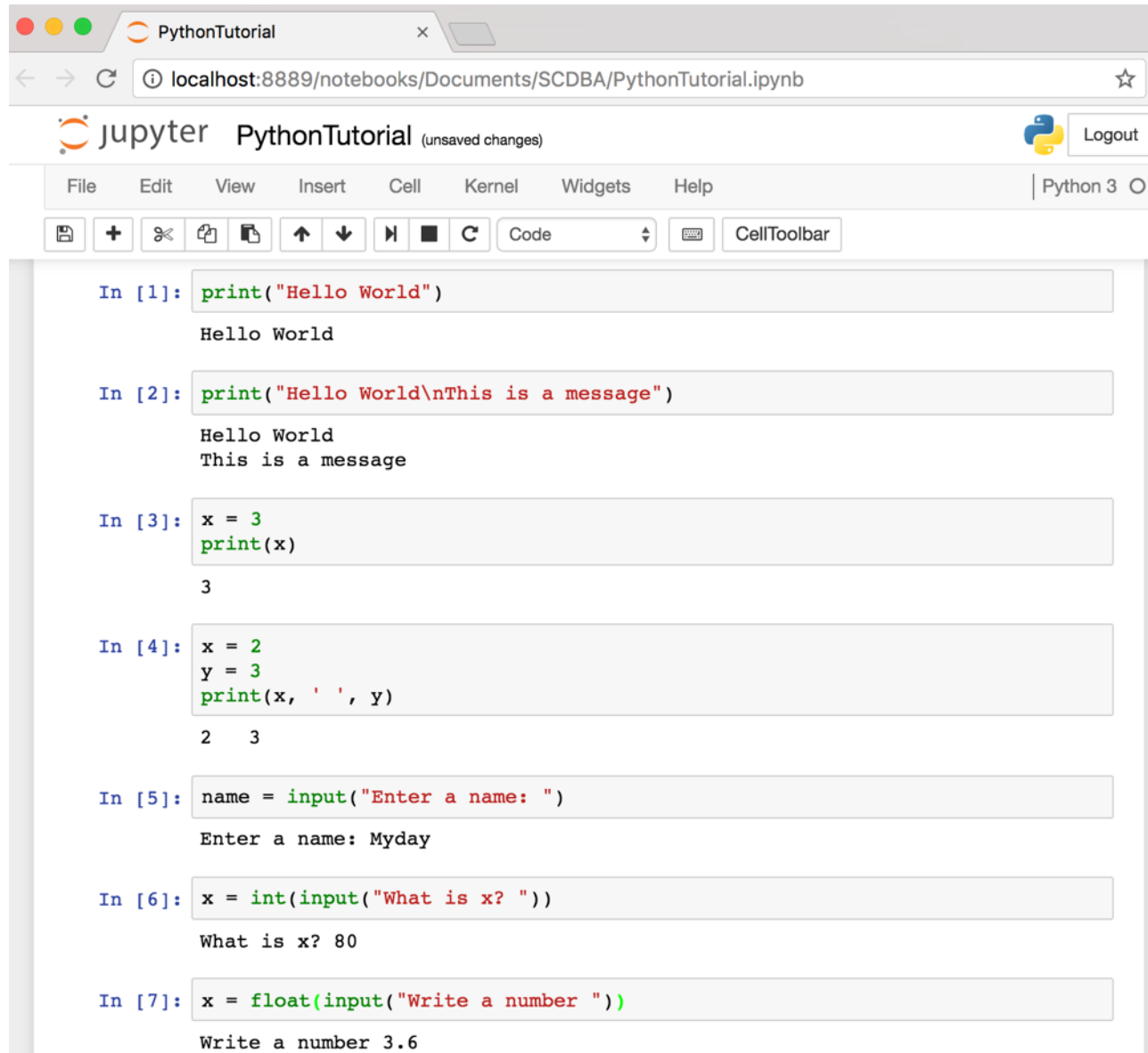
```
1 # https://www.learnpython.org/en/
2 # LearnPython.org interactive Python tutorial
3 print("Hello World")
4 print("Hello World\nThis is a message")
5 x = 3
6 print(x)
7 x = 2
8 y = 3
9 print(x, ' ', y).
```

 Output: `Hello World
Hello World
This is a message
3
2 3`
- Cell 4:**

```
1 # Python Variables
2 x = 2
3 price = 2.5
4 word = 'Hello'
5
6 word = 'Hello'
7 word = "Hello"
8 word = '''Hello'''
```

<https://tinyurl.com/aintpupython101>

# Text input and output



The screenshot shows a Jupyter Notebook window titled "PythonTutorial" with the URL `localhost:8889/notebooks/Documents/SCDBA/PythonTutorial.ipynb`. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations and execution. The notebook contains seven code cells, each with its input and output:

```
In [1]: print("Hello World")
Hello World

In [2]: print("Hello World\nThis is a message")
Hello World
This is a message

In [3]: x = 3
print(x)
3

In [4]: x = 2
y = 3
print(x, ' ', y)
2  3

In [5]: name = input("Enter a name: ")
Enter a name: Myday

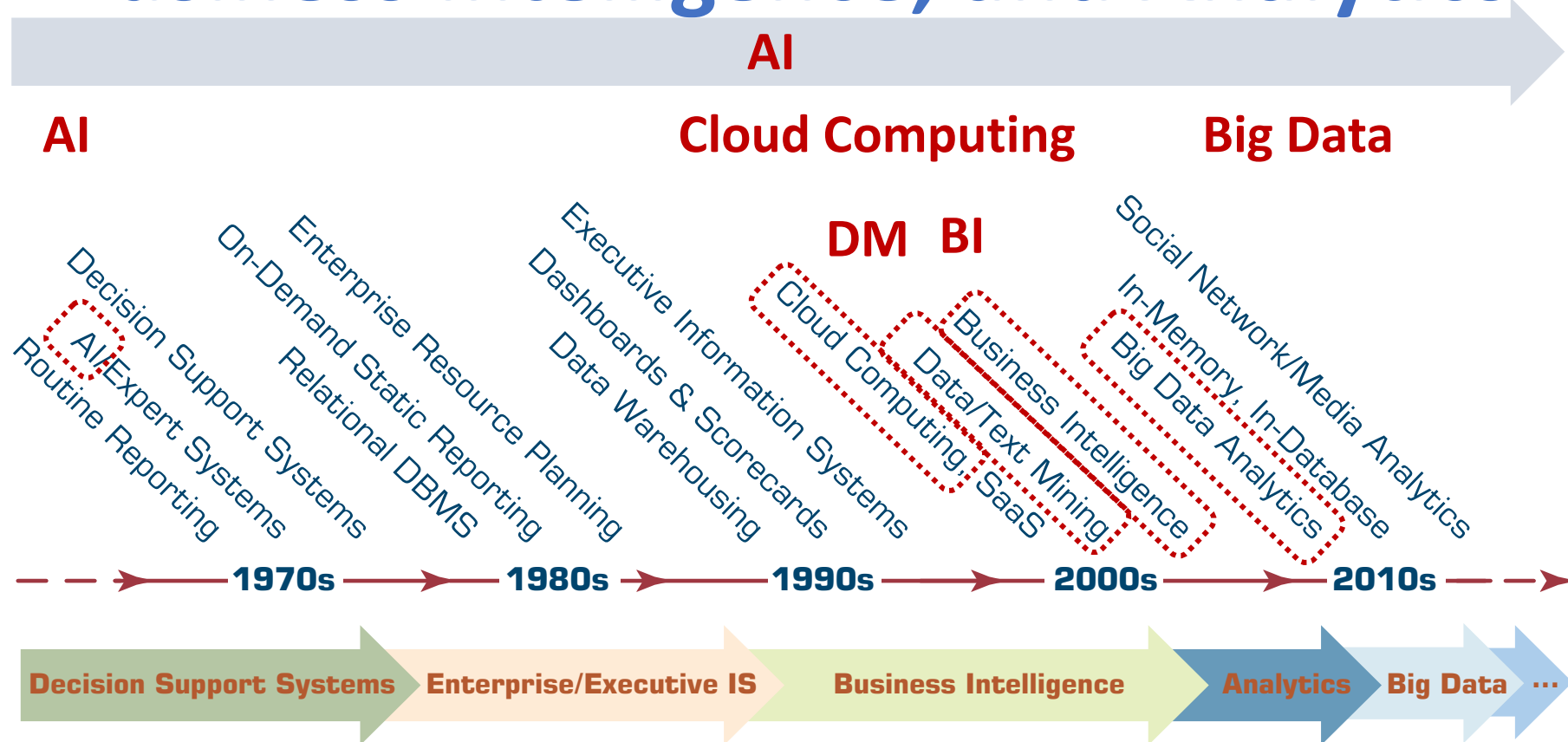
In [6]: x = int(input("What is x? "))
What is x? 80

In [7]: x = float(input("Write a number "))
Write a number 3.6
```

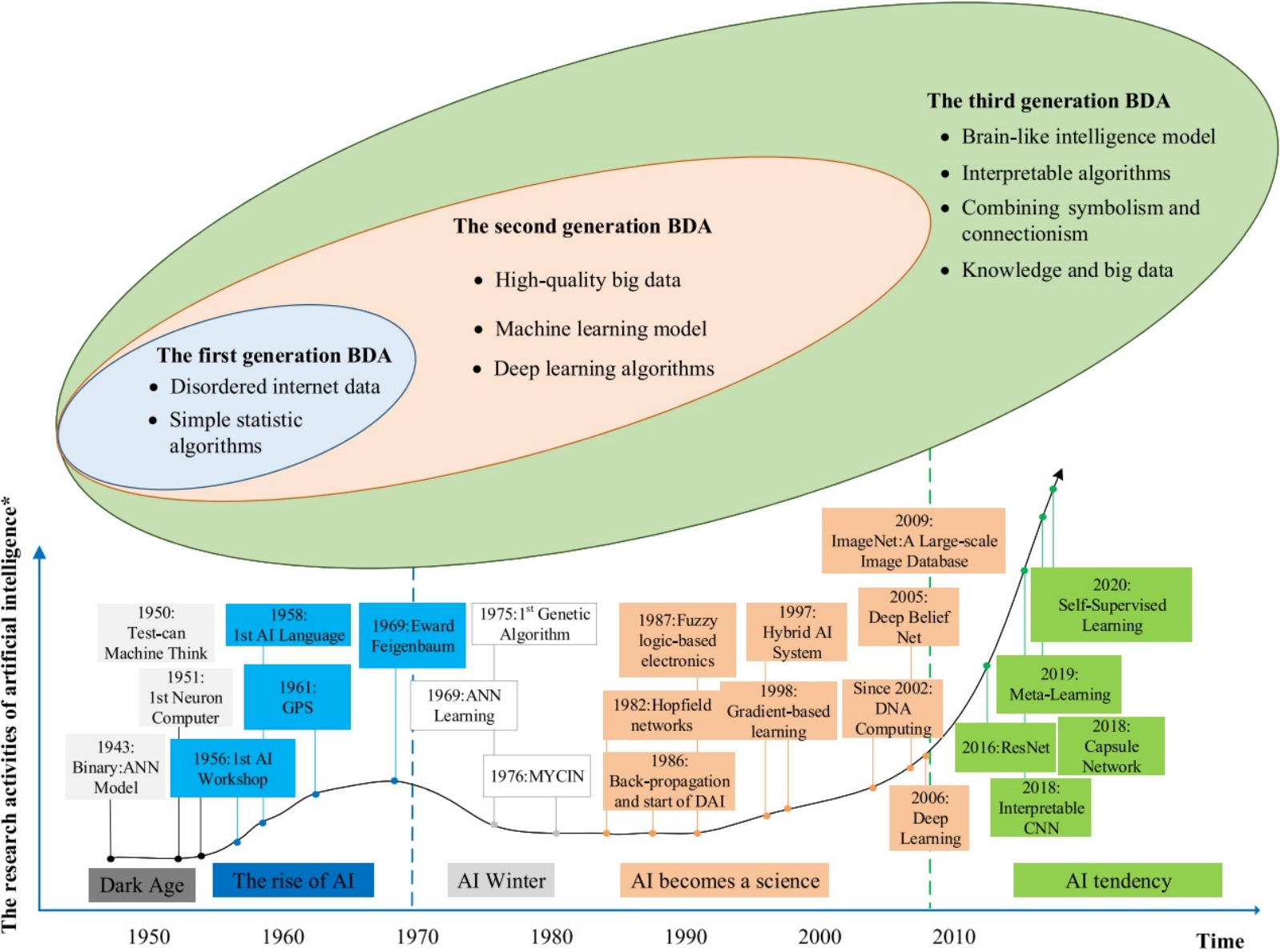
# Data Science

# AI, Big Data, Cloud Computing

## Evolution of Decision Support, Business Intelligence, and Analytics



# The Development of Big Data Analytics



Source: Wang, Junliang, Chuqiao Xu, Jie Zhang, and Ray Zhong (2022). "Big data analytics for intelligent manufacturing systems: A review." Journal of Manufacturing Systems 62 (2022): 738-752.

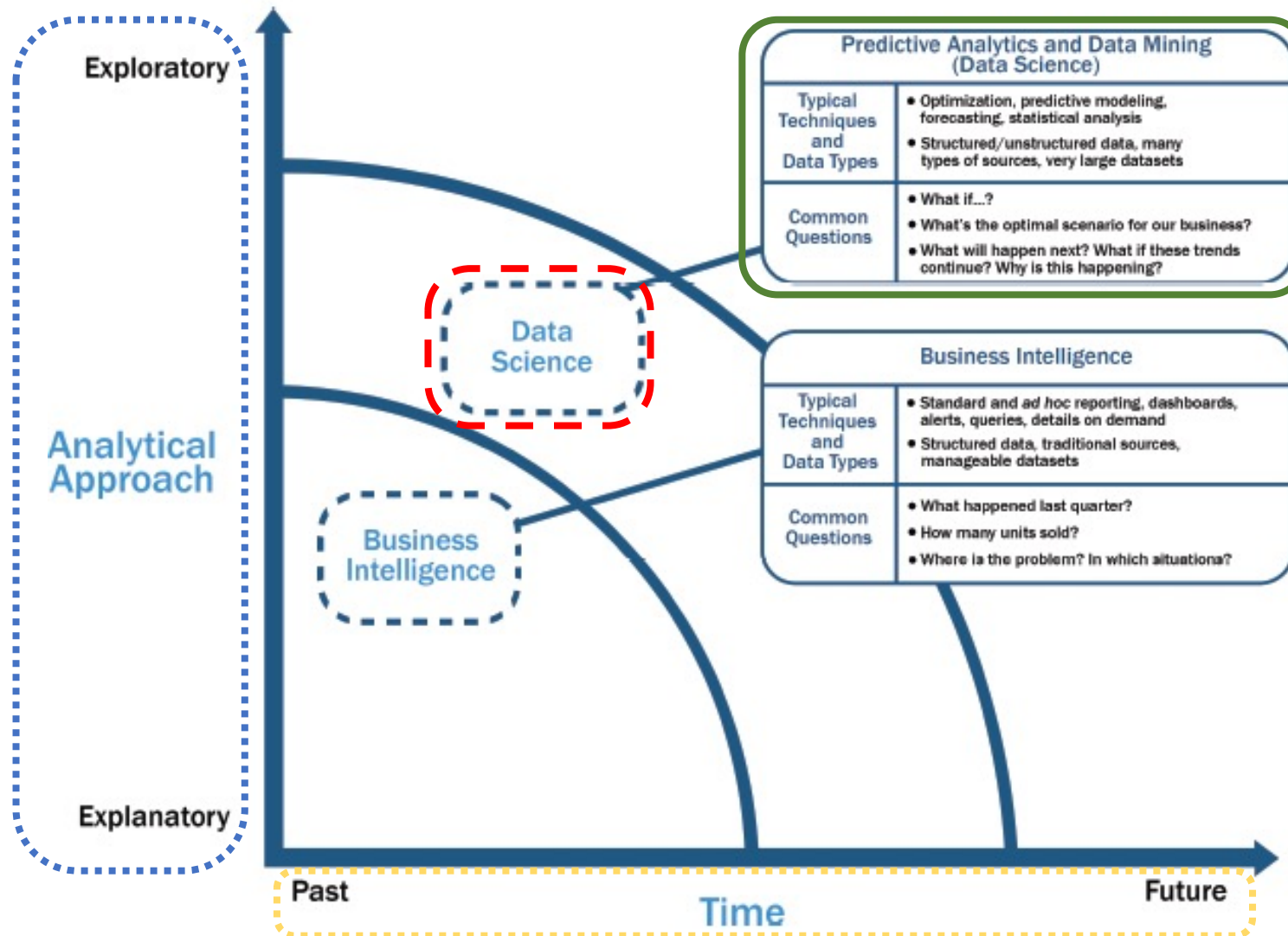
# Data Analyst

- Data analyst is just another term for professionals who were doing **BI** in the form of **data compilation, cleaning, reporting**, and perhaps some **visualization**.
- Their skill sets included Excel, some SQL knowledge, and reporting.
- You would recognize those capabilities as **descriptive or reporting analytics**.

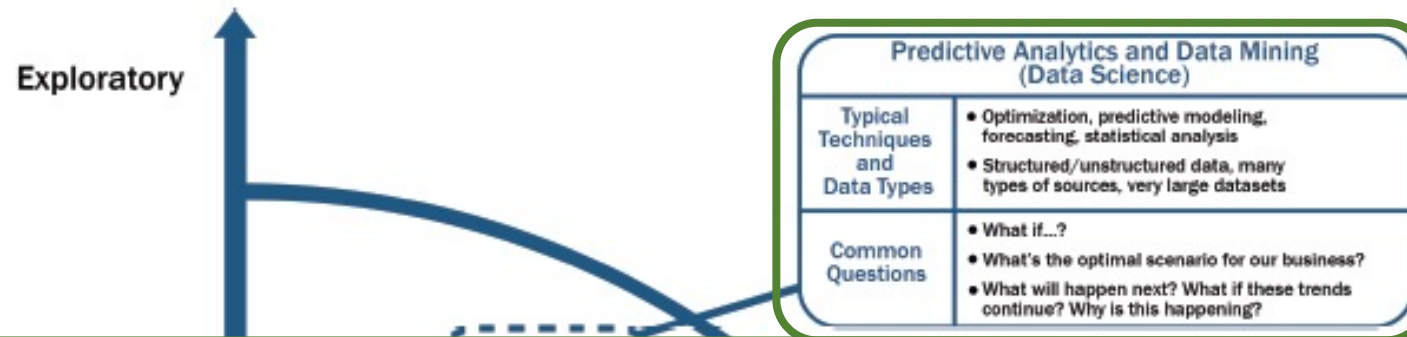
# Data Scientist

- Data scientist is responsible for **predictive analysis, statistical analysis,** and more **advanced analytical tools and algorithms.**
- They may have a deeper knowledge of algorithms and may recognize them under various labels—**data mining, knowledge discovery, or machine learning.**
- Some of these professionals may also need deeper programming knowledge to be able to write code for data cleaning/analysis in current Web-oriented languages such as Java or Python and statistical languages such as R.
- Many analytics professionals also need to build significant expertise in **statistical modeling, experimentation, and analysis.**

# Data Science and Business Intelligence



# Data Science and Business Intelligence



## Predictive Analytics and Data Mining (Data Science)

Past

Time

Future

# Predictive Analytics and Data Mining (Data Science)

Structured/unstructured data, many types of sources,  
very large datasets

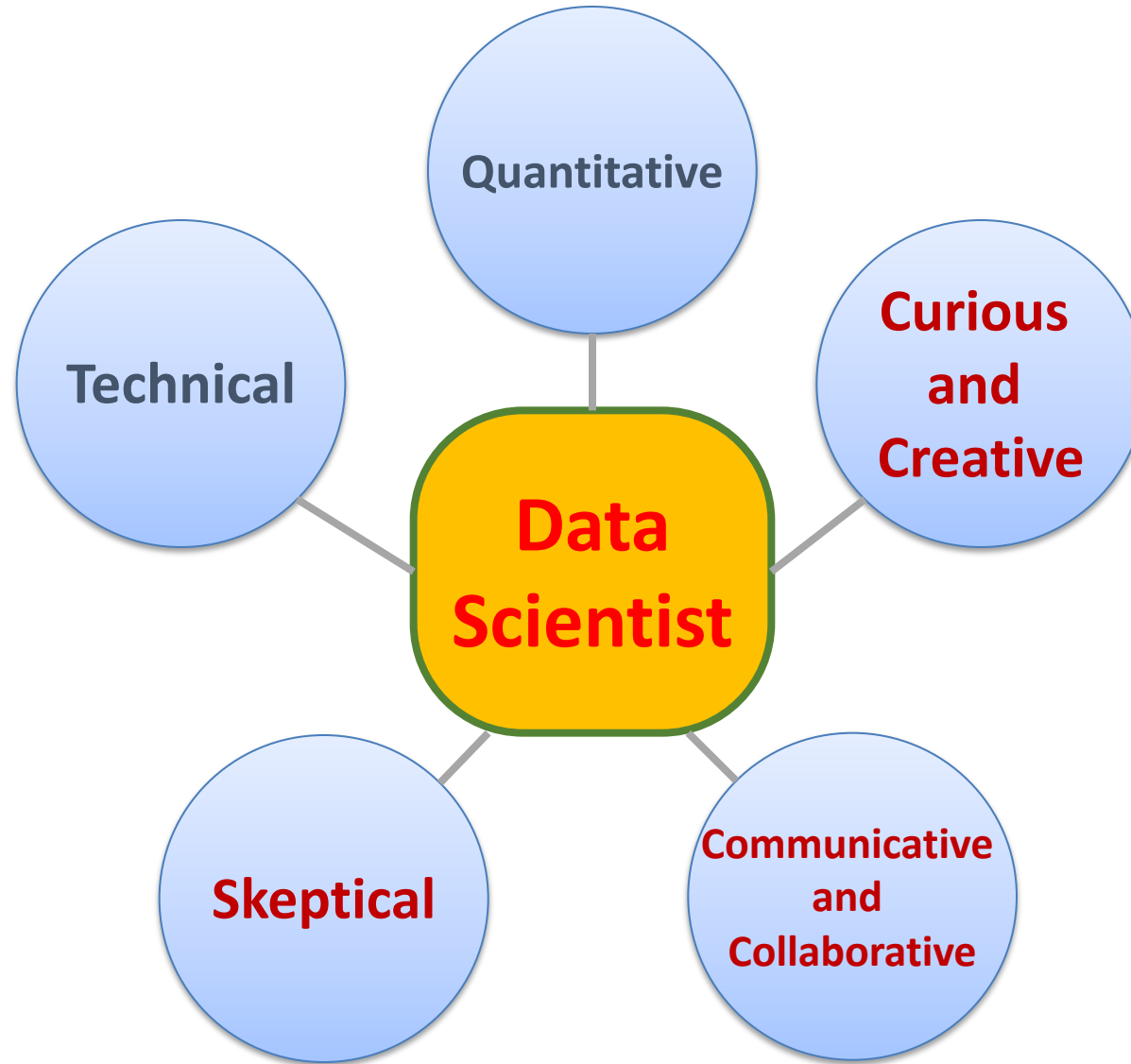
Optimization, predictive modeling, forecasting statistical analysis

What if...?  
What's the optimal scenario for our business?  
What will happen next?  
What if these trends continue?  
Why is this happening?

# Profile of a Data Scientist

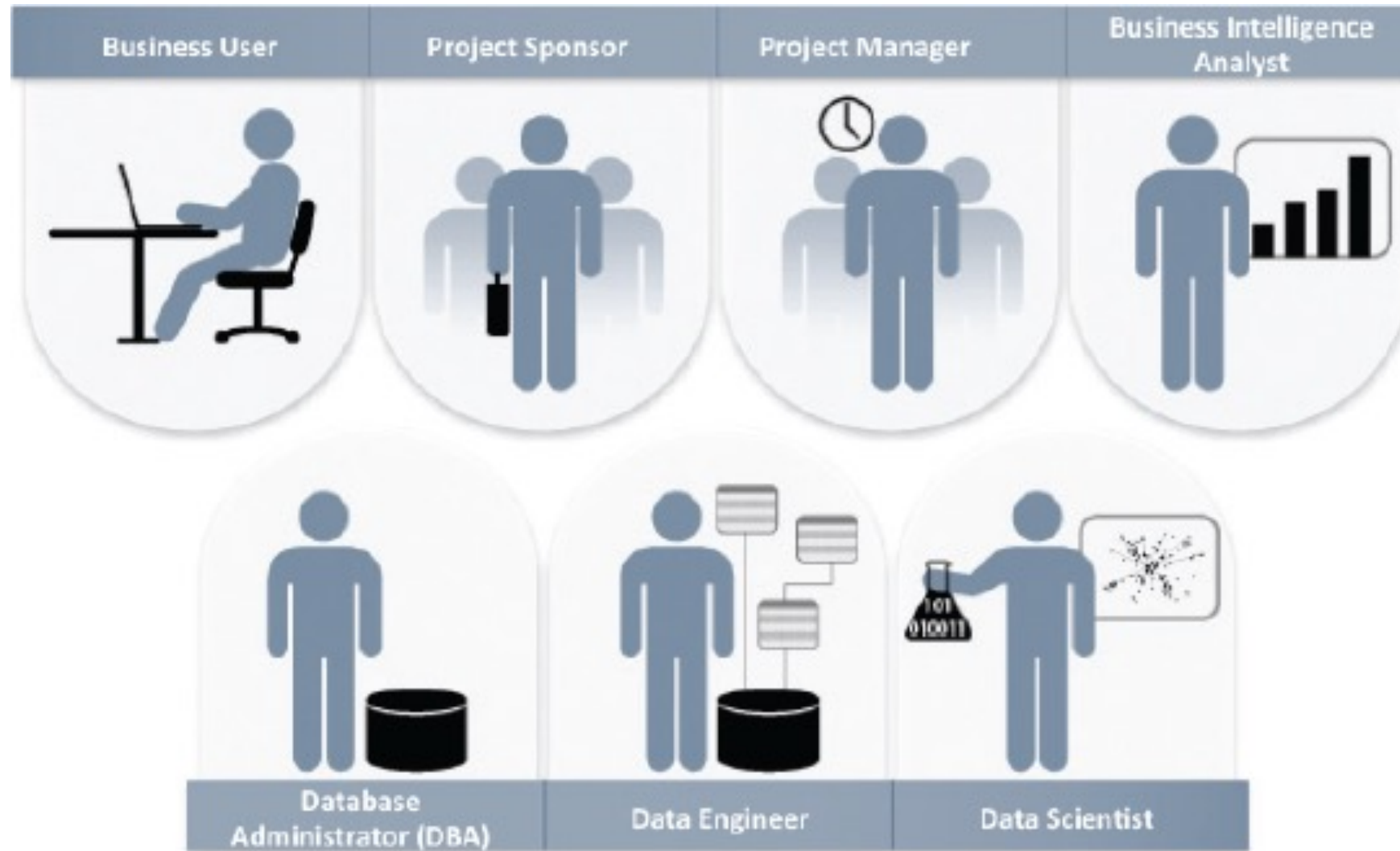
- **Quantitative**
  - **mathematics or statistics**
- **Technical**
  - **software engineering, machine learning, and programming skills**
- **Skeptical mind-set and critical thinking**
- **Curious and creative**
- **Communicative and collaborative**

# Data Scientist Profile

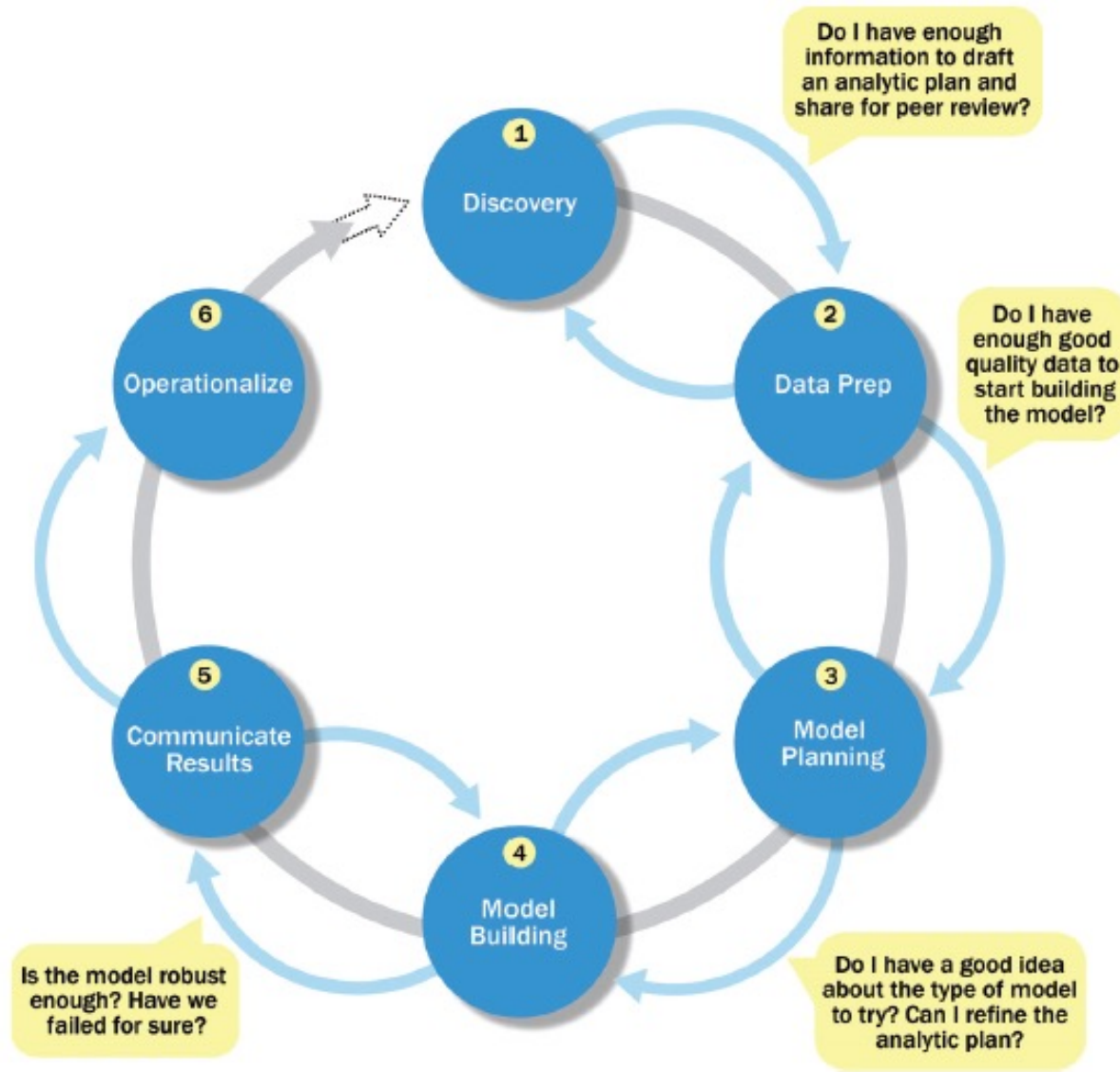


# Big Data Analytics Lifecycle

# Key Roles for a Successful Analytics Project



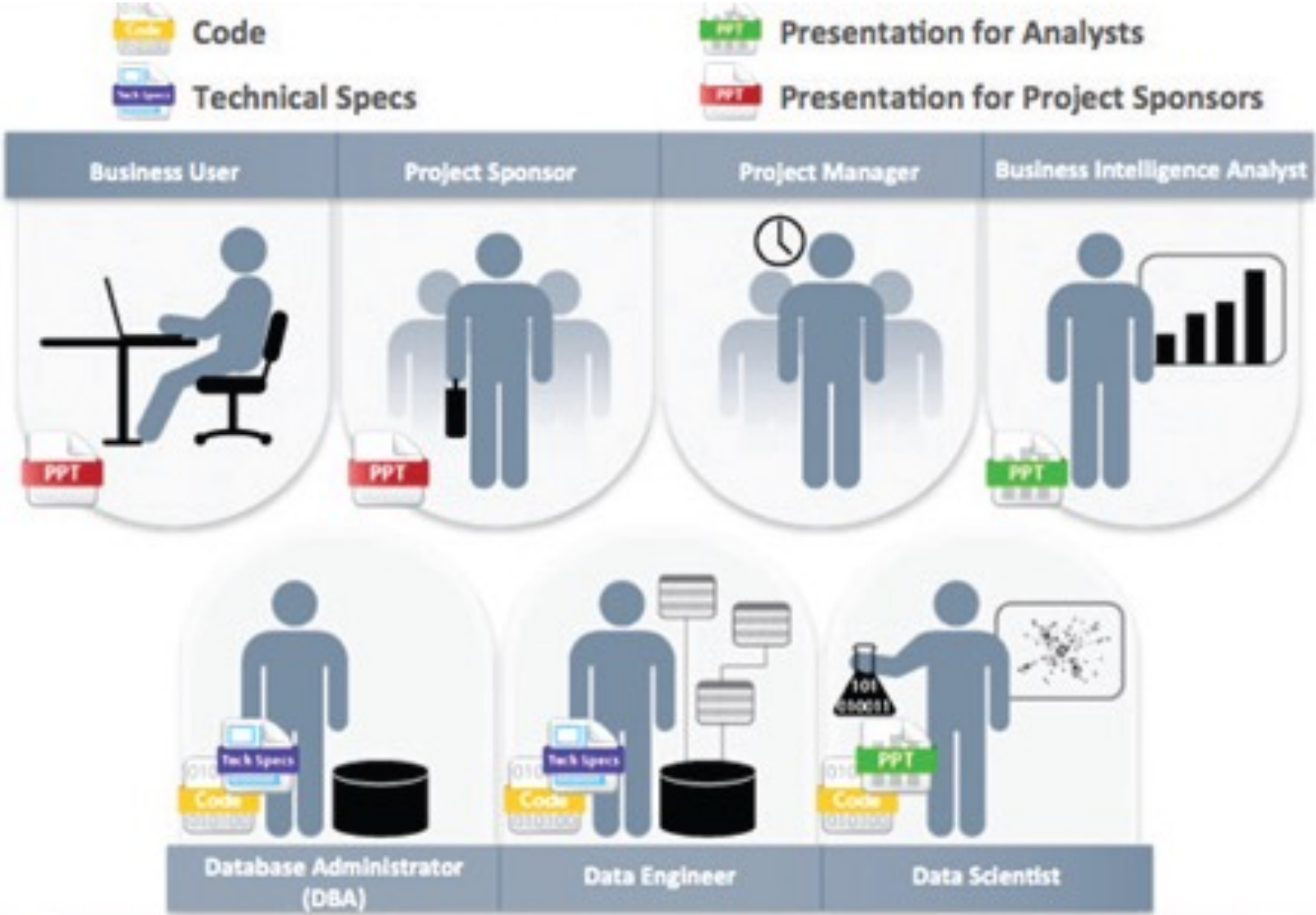
# Overview of Data Analytics Lifecycle



# Overview of Data Analytics Lifecycle

- 1. Discovery**
- 2. Data preparation**
- 3. Model planning**
- 4. Model building**
- 5. Communicate results**
- 6. Operationalize**

# Key Outputs from a Successful Analytics Project

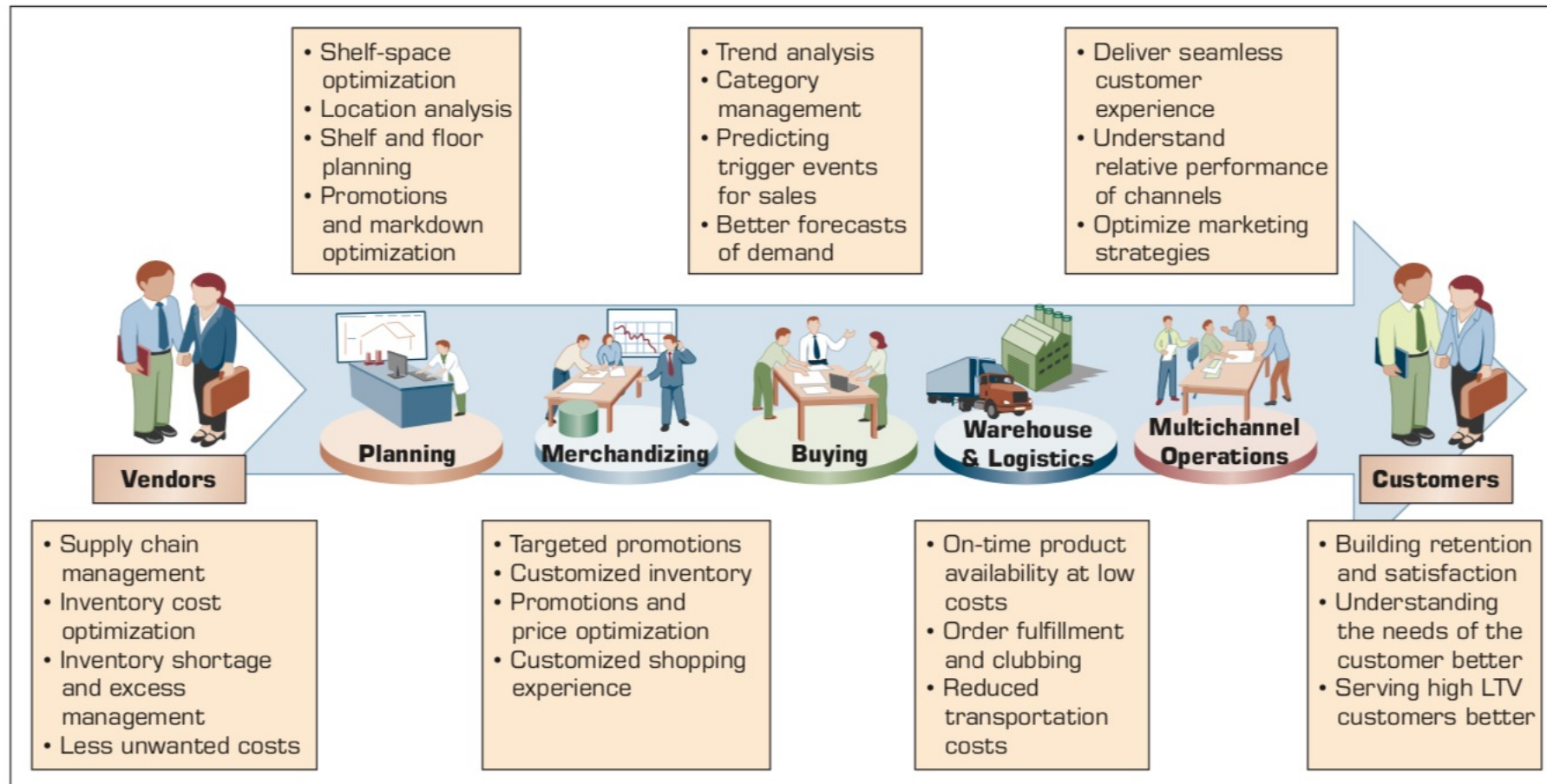


Source: EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, Wiley, 2015

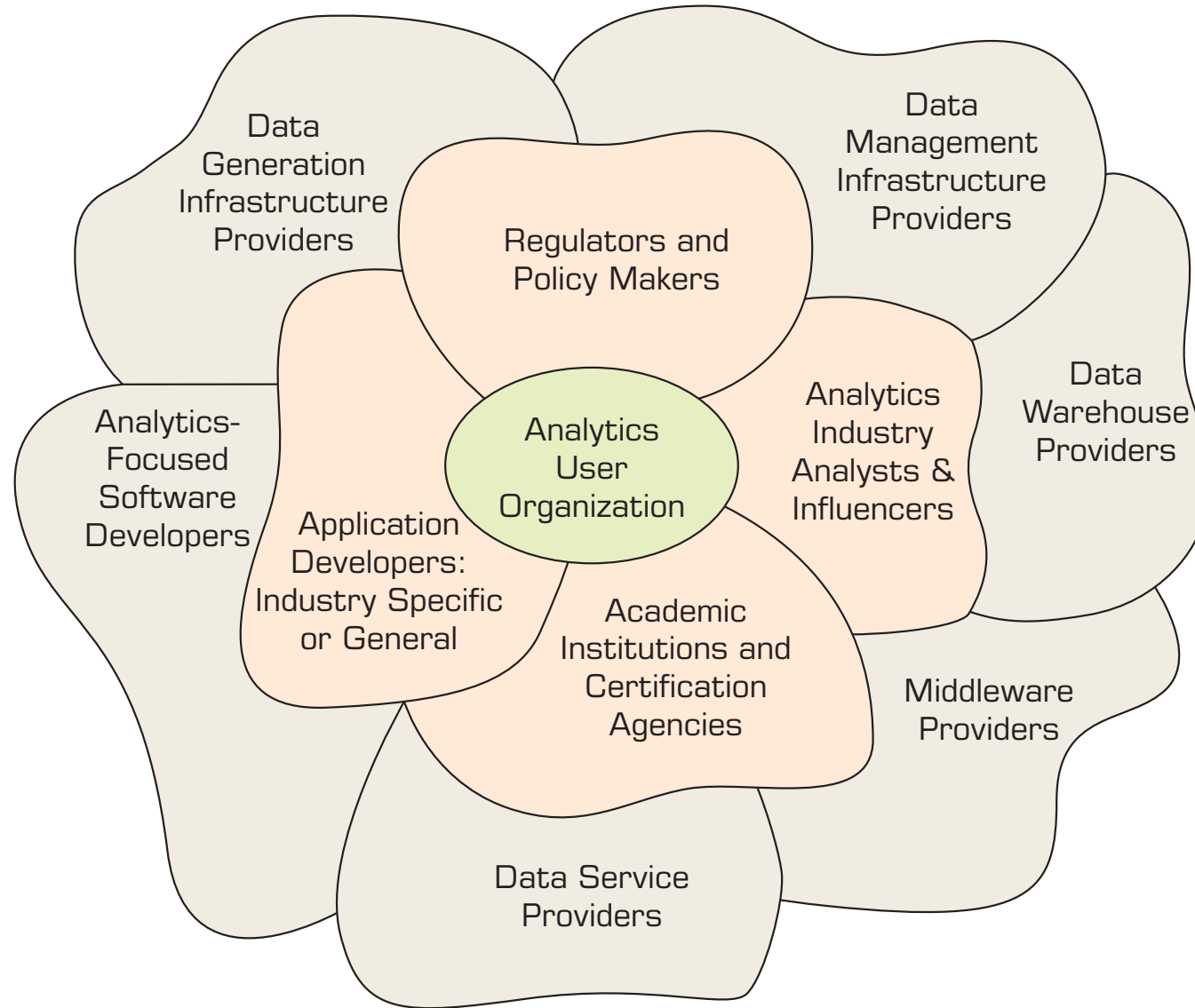
# Example of Analytics Applications in a Retail Value Chain

## Retail Value Chain

Critical needs at every touch point of the Retail Value Chain

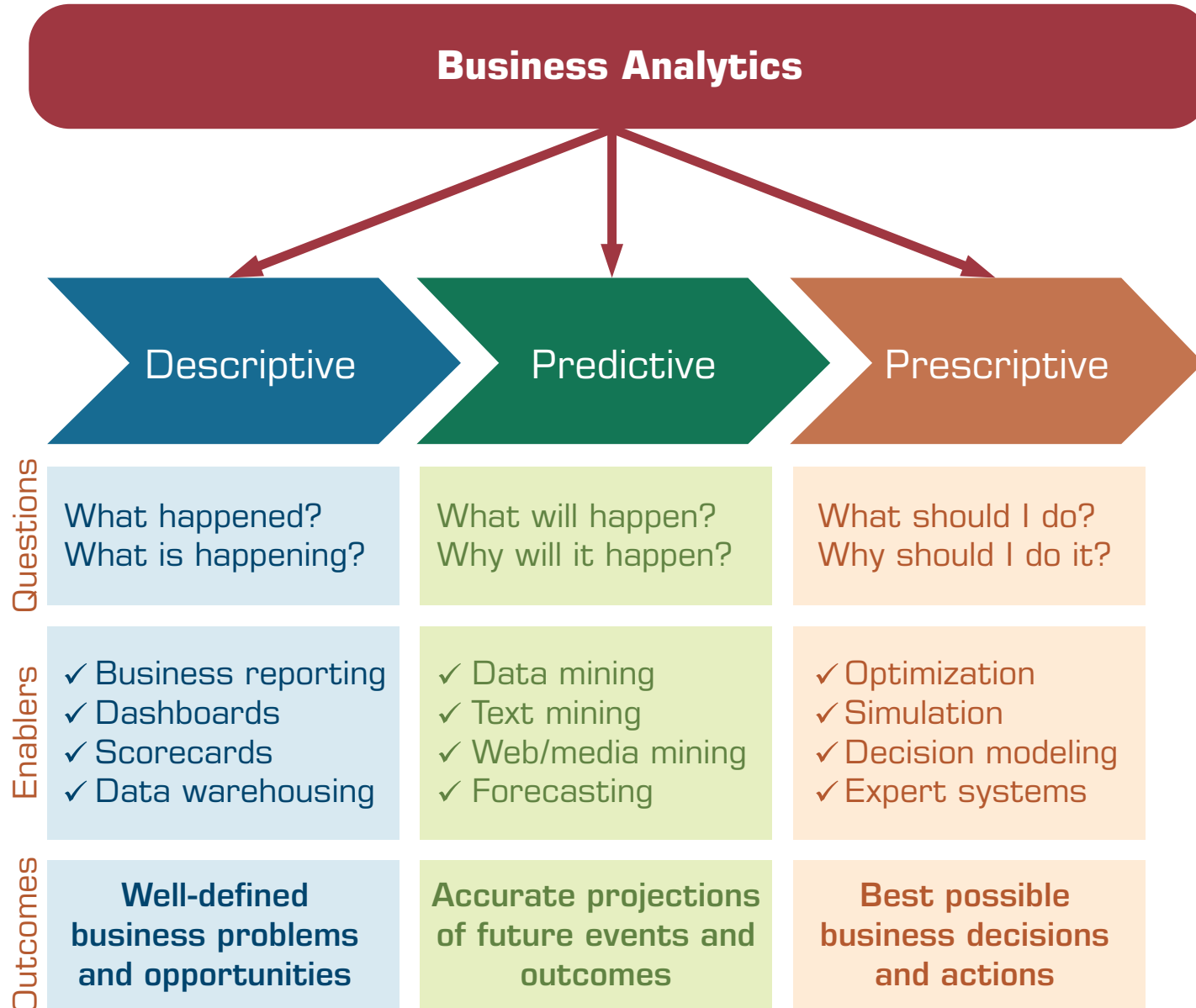


# Analytics Ecosystem

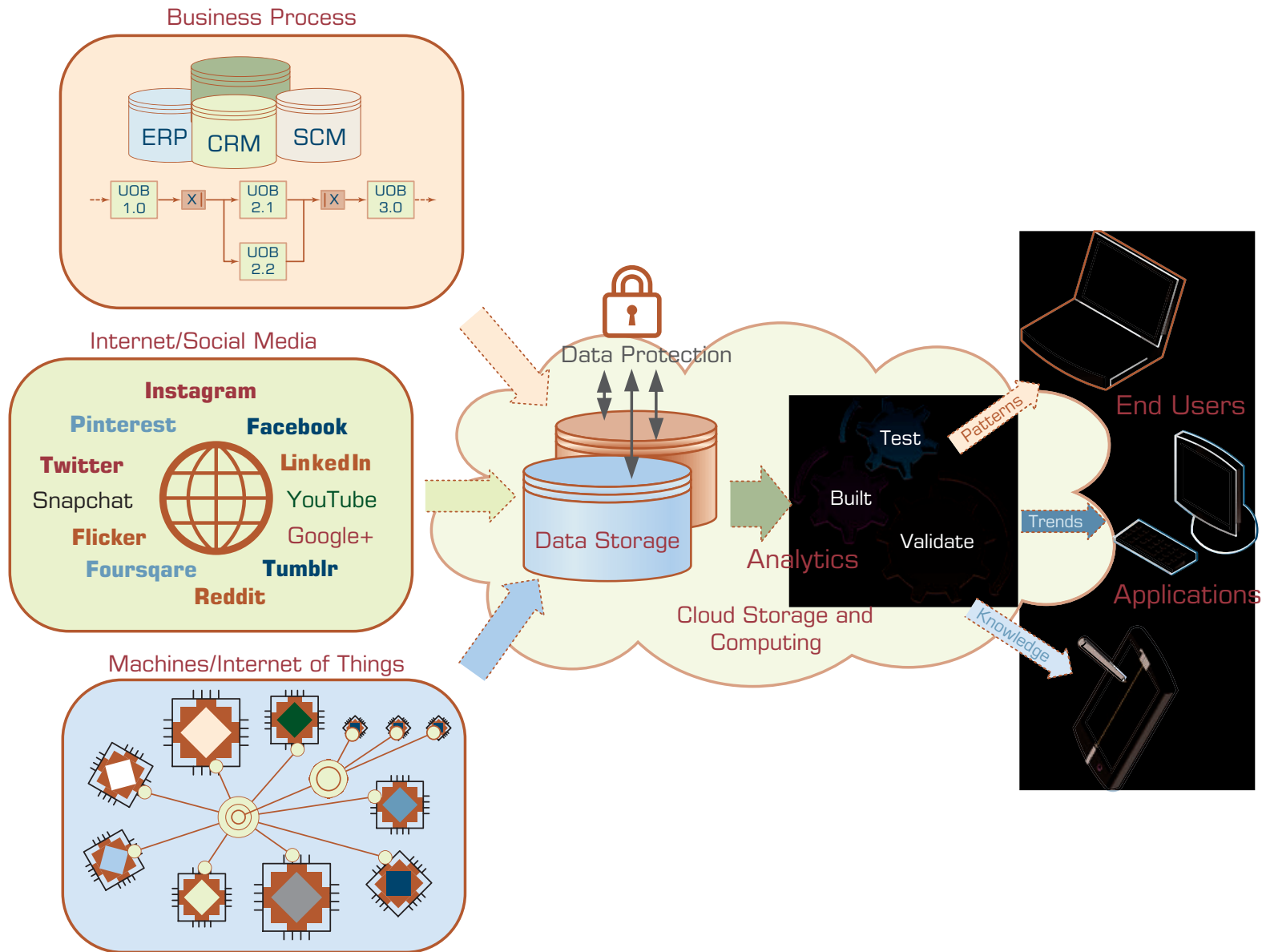




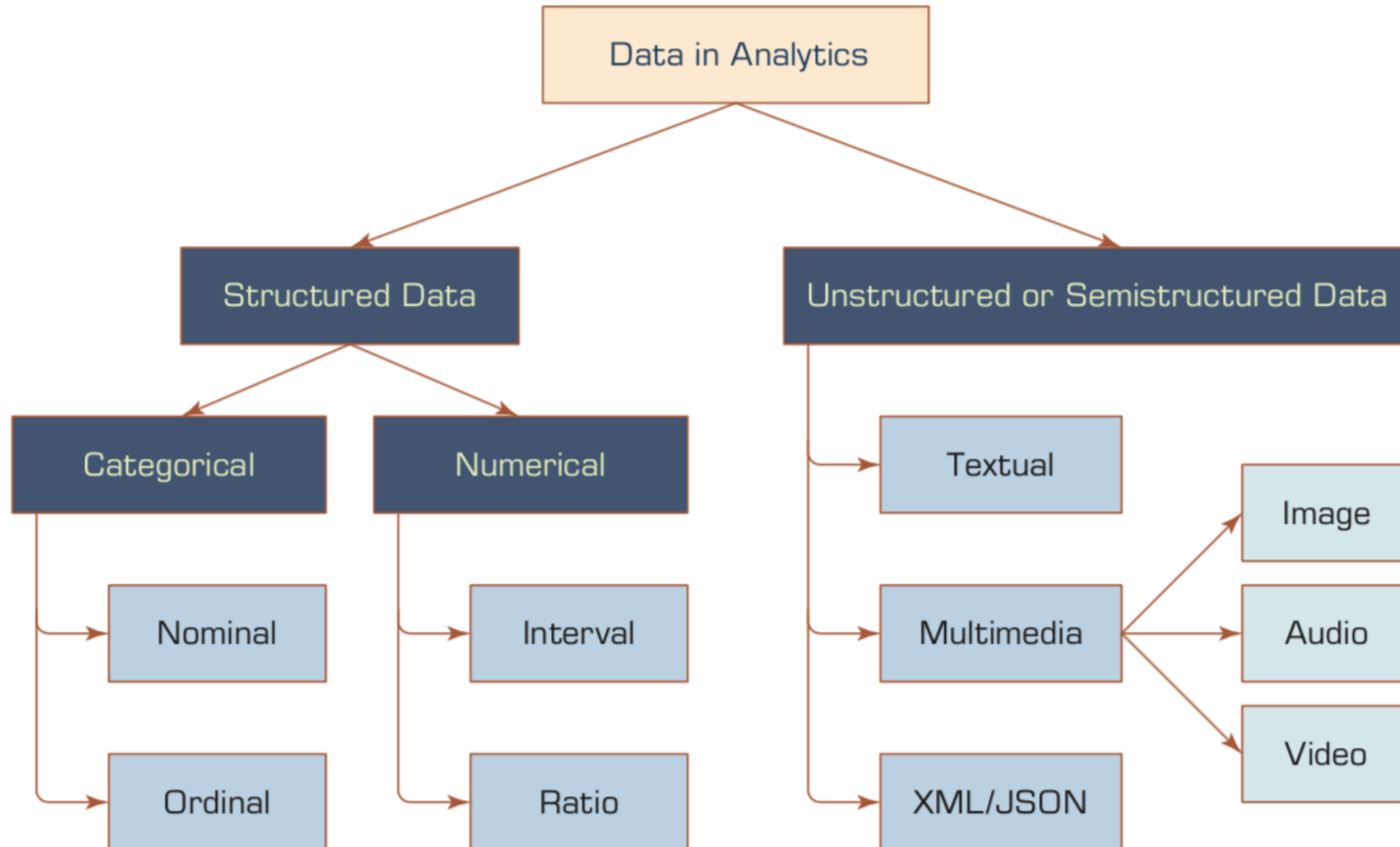
# Three Types of Analytics



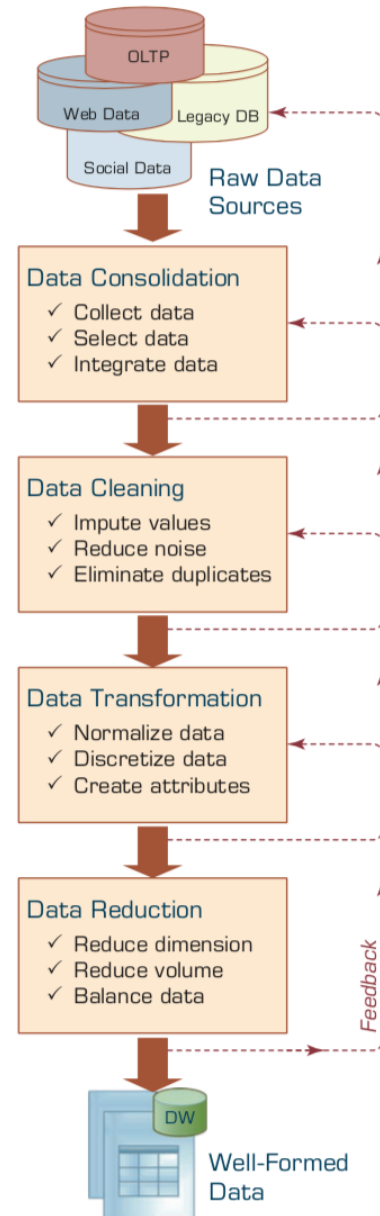
# A Data to Knowledge Continuum



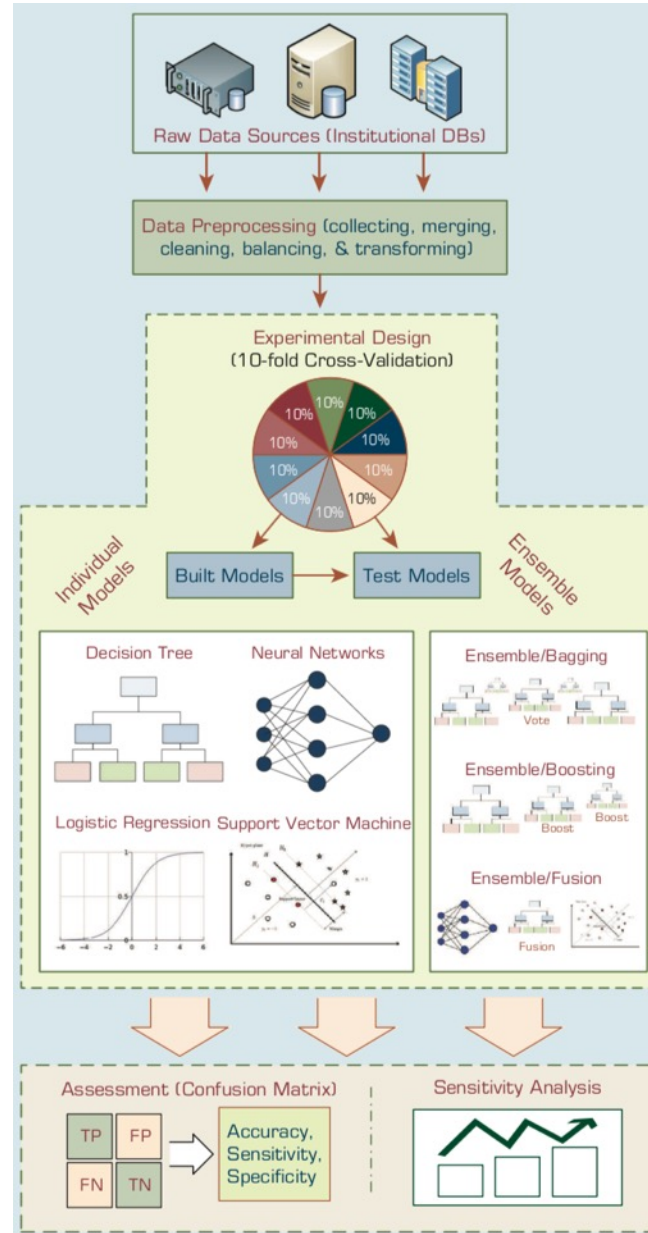
# A Simple Taxonomy of Data



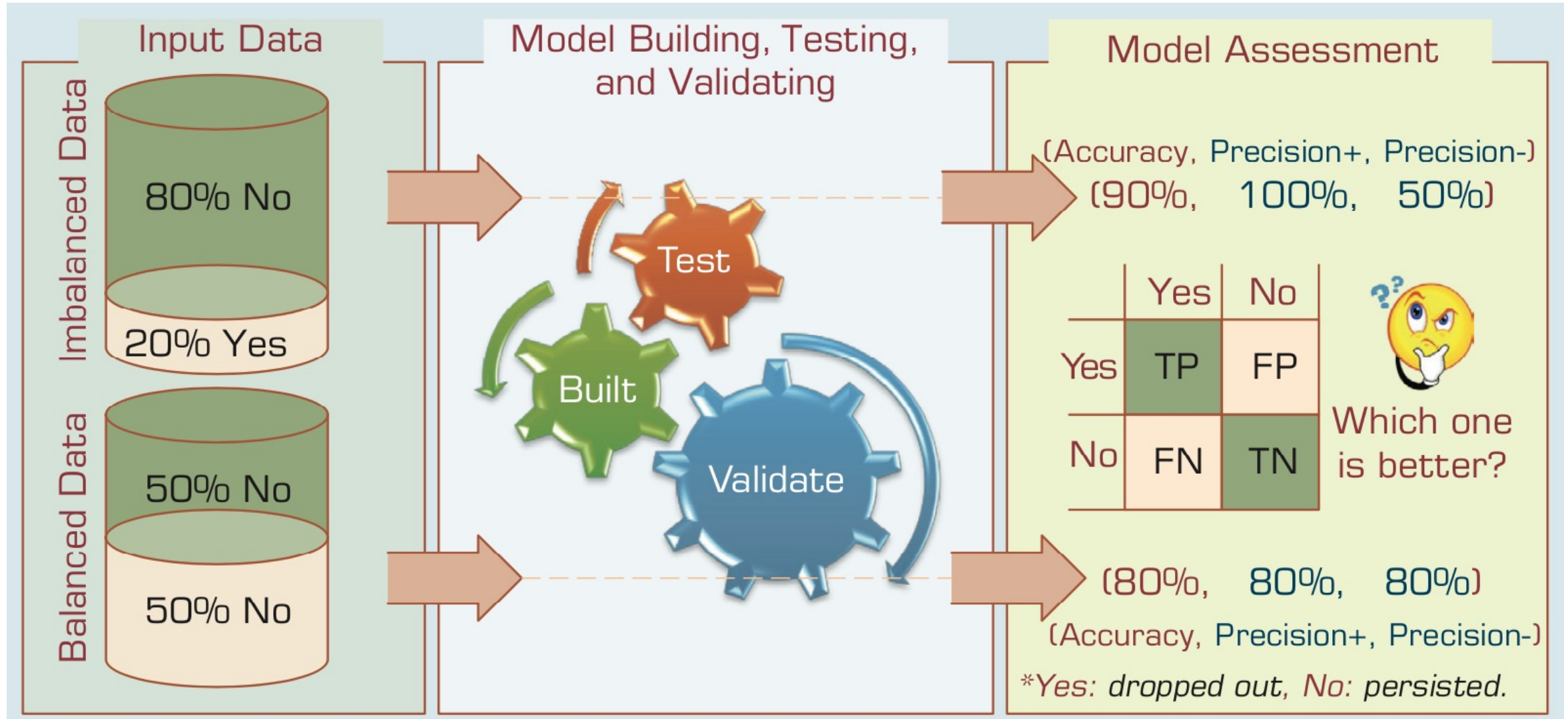
# Data Preprocessing Steps



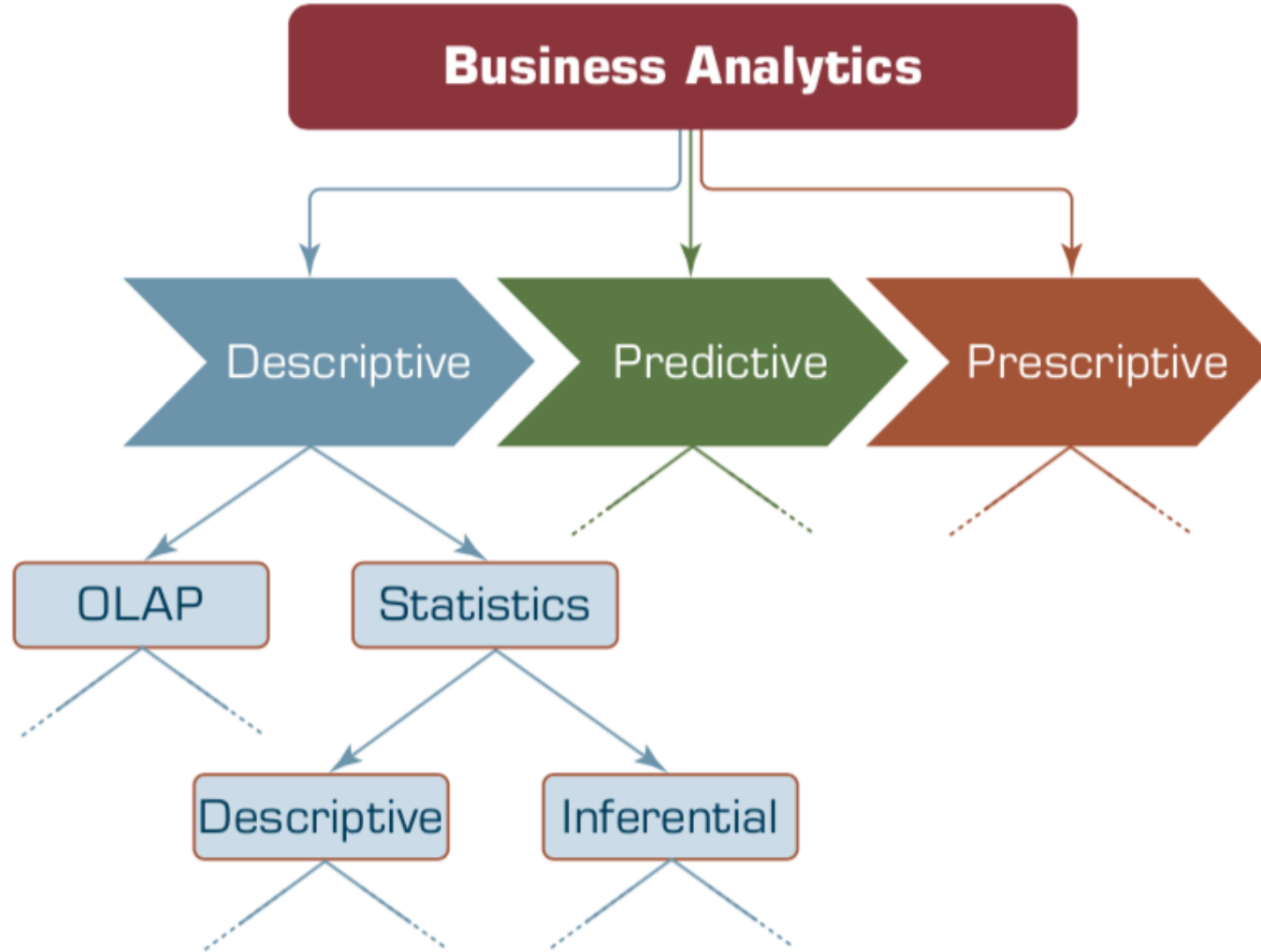
# An Analytics Approach to Predicting Student Attrition



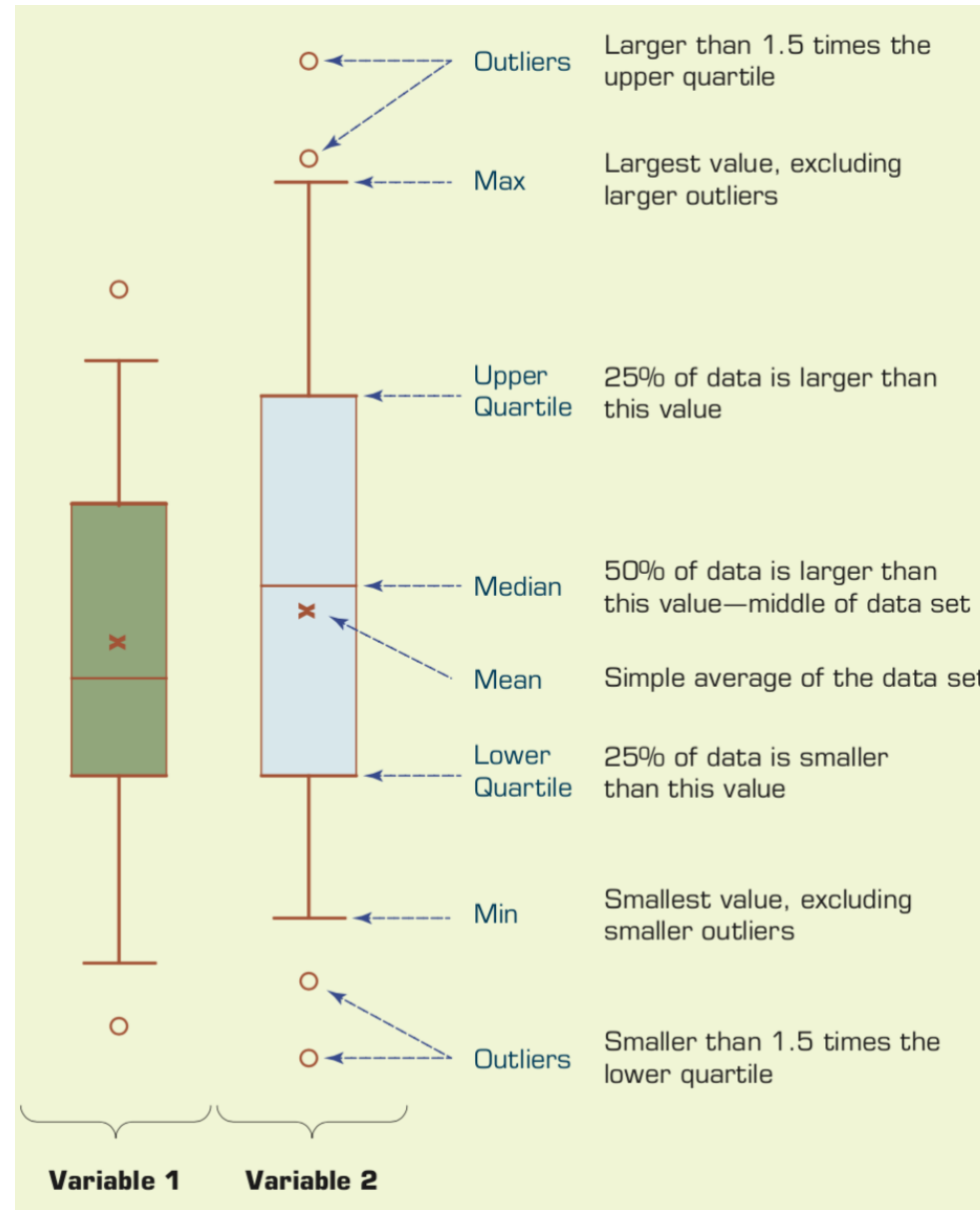
# A Graphical Depiction of the Class Imbalance Problem



# Relationship between Statistics and Descriptive Analytics

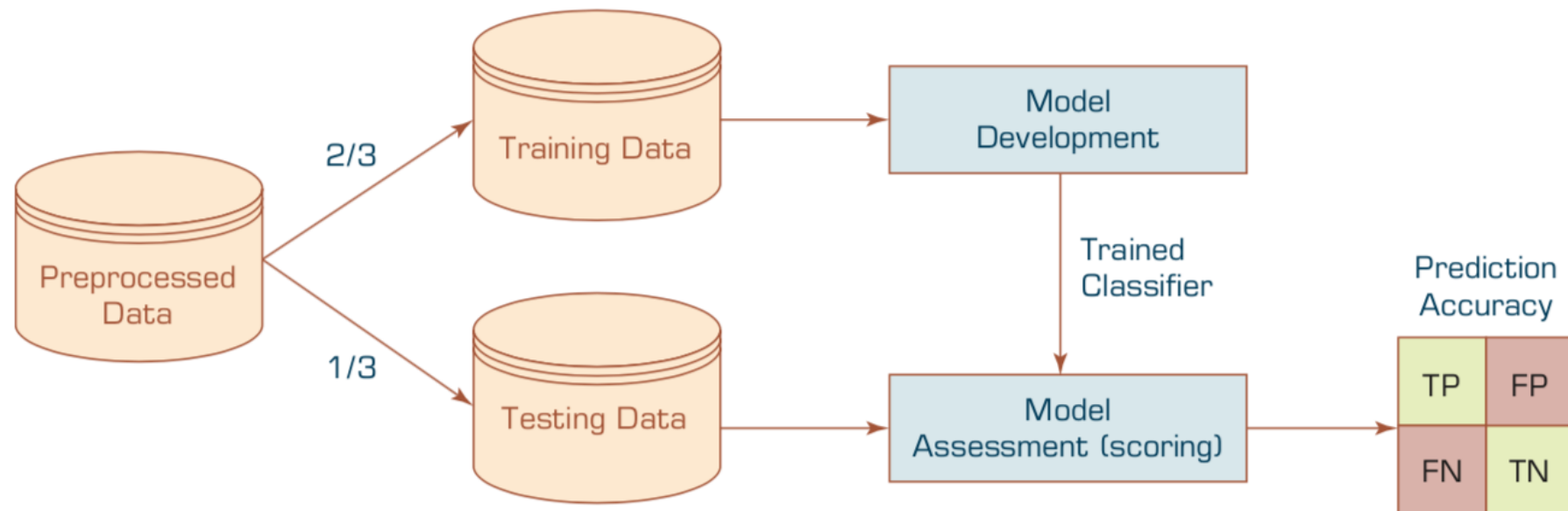


# Understanding the Specifics about Box-and-Whiskers Plots



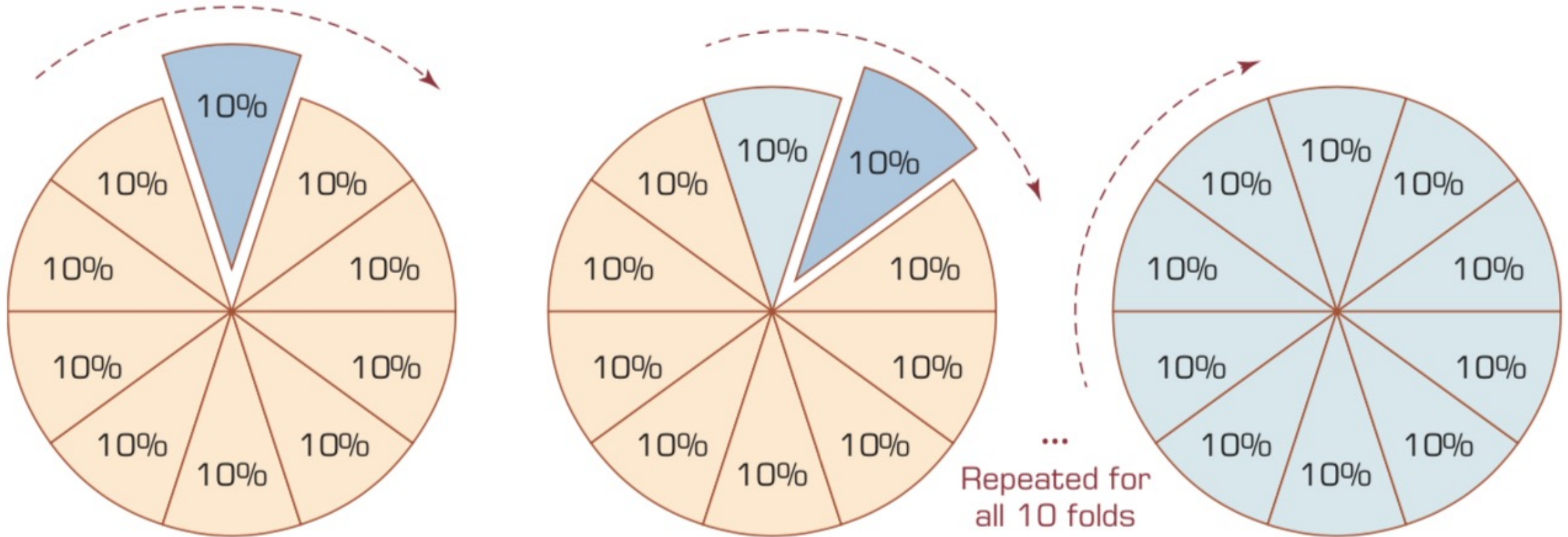
# Estimation Methodologies for Machine Learning Classification Tasks

- **Simple split** (or holdout or test sample estimation)
  - Split the data into 2 mutually exclusive sets training (~70%) and testing (30%)



- For ANN, the data is split into three sub-sets (training [~60%], validation [~20%], testing [~20%])

# *k*-Fold Cross-Validation: Machine Learning Data Splitting and Model Assessment



# **Data Science for Sustainability and ESG**

# Data Science for Sustainability and ESG

- **Data Science and Sustainability**
- **Data Collection and Analysis for Sustainability**
- **Implementing Data-Driven Sustainability Strategies**

# Fundamentals of Data Science

- **Definition of Data Science:**
  - Interdisciplinary field using **scientific methods, processes, algorithms and systems** to **extract knowledge and insights** from **structured and unstructured data**.
- **Key components:**
  - **Statistics, Machine Learning, Data Engineering**

# Data Science Process

- 1. Business Understanding**
- 2. Data Understanding**
- 3. Data Preparation**
- 4. Modeling**
- 5. Evaluation**
- 6. Deployment**

# Sustainability and ESG

- **Sustainability: Meeting present needs without compromising future generations**
- **ESG: Environmental, Social, and Governance**
  - **Environmental: Climate change, resource depletion, waste, pollution, deforestation**
  - **Social: Human rights, labor standards, workplace safety, community relations**
  - **Governance: Board diversity, executive compensation, ethics, transparency**

# Sustainability and ESG: Business case

- **Risk management**
- **Cost savings**
- **Innovation**
- **Brand value**
- **Investor attraction**

# Intersection of Data Science with Sustainability and ESG

- **Data-driven decision making for sustainability initiatives**
- **Predictive analytics for environmental impact**
- **Machine learning for optimizing resource usage**
- **Big data analysis for social impact assessment**
- **AI-powered governance risk management**

# The Interconnectedness of Data, Sustainability & ESG

- **Sustainability encompasses environmental, social, and governance concerns.**
- **ESG provides a framework for measuring and reporting sustainability performance.**
- **Data science offers the tools to collect, analyze, and use ESG data for decision-making.**

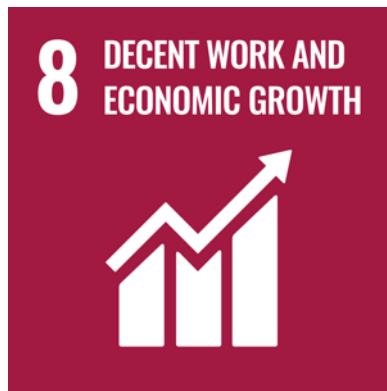
**ESG:**

**Environmental**

**Social**

**Governance**

# Sustainable Development Goals (SDGs)



# Sustainable Development Goals (SDGs) and 5P

Partnership

Peace

Prosperity

People

Planet



Source: Folke, Carl, Reinette Biggs, Albert V. Norström, Belinda Reyers, and Johan Rockström. "Social-ecological resilience and biosphere-based sustainability science." Ecology and Society 21, no. 3 (2016).

# ESG to 17 SDGs

## ENVIRONMENT



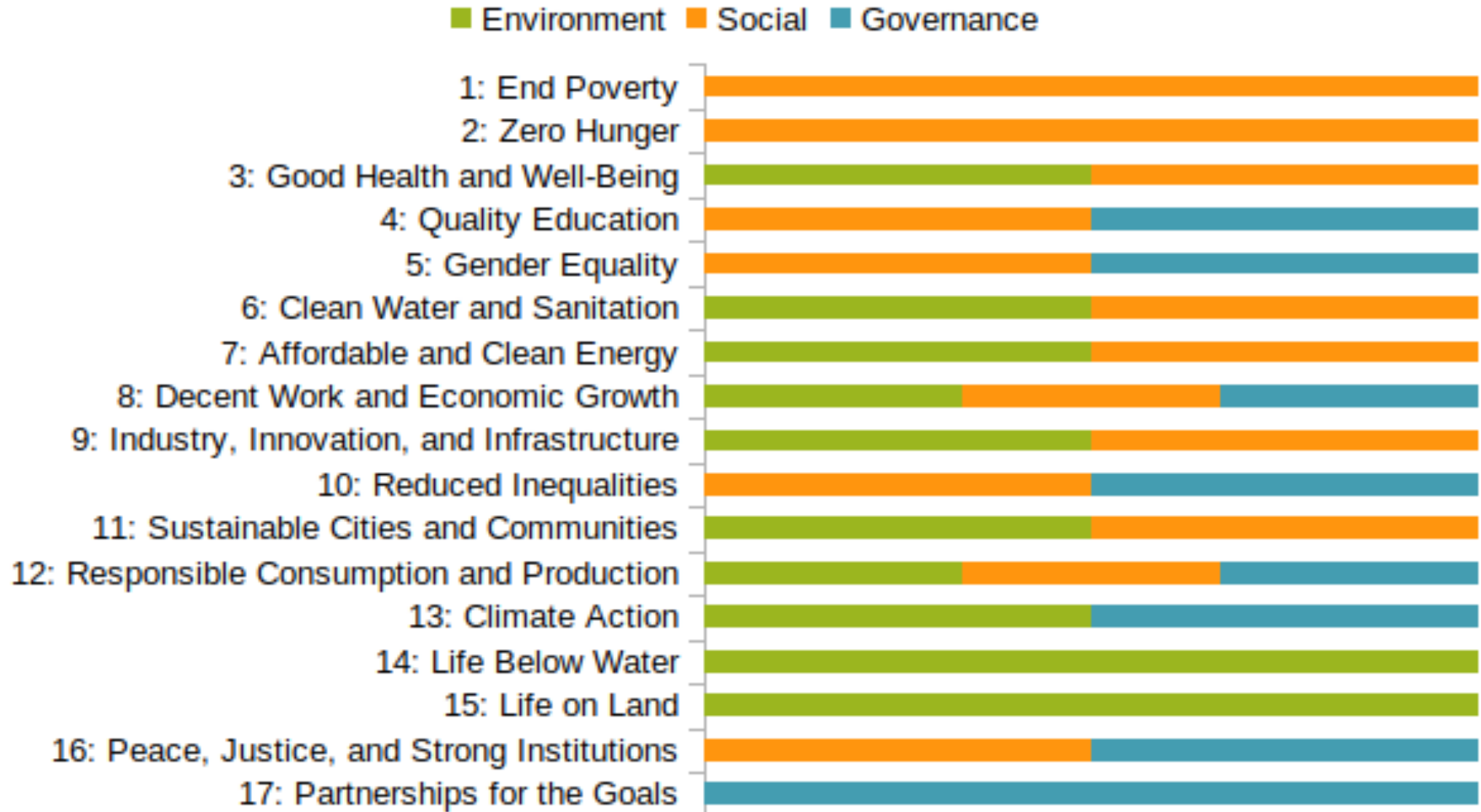
## SOCIAL



## GOVERNANCE



# ESG to 17 SDGs



# ESG Reporting Frameworks and Standards

1. **GRI (Global Report Initiative):**  
<https://www.globalreporting.org/>
2. **CDP (Carbon Disclosure Project):**  
<https://www.cdp.net/>
3. **SASB (Sustainability Accounting Standards Board):**  
<https://sasb.org/>
4. **ISSB (International Sustainability Standards Board):**  
<https://www.ifrs.org/groups/international-sustainability-standards-board/>
5. **TCFD (Task Force on Climate-related Financial Disclosures):**  
<https://www.fsb-tcf.org/>

# Summary

- **Python Programming**
- **Data Science**

# References

- Wes McKinney (2022), "Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter", 3rd Edition, O'Reilly Media.
- Aurélien Géron (2023), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 3rd Edition, O'Reilly Media.
- Steven D'Ascoli (2022), Artificial Intelligence and Deep Learning with Python: Every Line of Code Explained For Readers New to AI and New to Python, Independently published.
- Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson.
- Denis Rothman (2024), Transformers for Natural Language Processing and Computer Vision - Third Edition: Explore Generative AI and Large Language Models with Hugging Face, ChatGPT, GPT-4V, and DALL-E 3, 3rd ed. Edition, Packt Publishing
- Ben Auffarth (2023), Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT and other LLMs, Packt Publishing.
- Varun Grover, Roger HL Chiang, Ting-Peng Liang, and Dongsong Zhang (2018), "Creating Strategic Business Value from Big Data Analytics: A Research Framework", Journal of Management Information Systems, 35, no. 2, pp. 388-423.
- Junliang Wang, Chuqiao Xu, Jie Zhang, and Ray Zhong (2022). "Big data analytics for intelligent manufacturing systems: A review." Journal of Manufacturing Systems 62 (2022): 738-752.
- Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson
- Thomas R. Caldwell (2025), The Agentic AI Bible: The Complete and Up-to-Date Guide to Design, Build, and Scale Goal-Driven, LLM-Powered Agents that Think, Execute and Evolve, Independently published.
- Python Programming, <https://pythonprogramming.net/>
- Python, <https://www.python.org/>
- Python Programming Language, <http://pythonprogramminglanguage.com/>
- Numpy, <http://www.numpy.org/>
- Pandas, <http://pandas.pydata.org/>
- Skikit-learn, <http://scikit-learn.org/>
- W3Schools Python, <https://www.w3schools.com/python/>
- Learn Python, <https://www.learnpython.org/>
- Google's Python Class, <https://developers.google.com/edu/python>
- Min-Yuh Day (2025), Python 101, <https://tinyurl.com/aintpupython101>