**Data 601:**

**Data Science**

**Quick Reference Guide**

**Felix Gonzalez, P.E.**

**Adjunct Instructor**

**Division of Professional Studies**

**Computer Science and Electrical Engineering**

**University of Maryland Baltimore County**

Table of Contents

[Data 601 Course Main Tools and Important References 1](#_Toc184118094)

[Python Data Science Packages and Libraries 2](#_Toc184118095)

[Referencem, Sites, and Examples 5](#_Toc184118096)

[AI Standards, And Ethics Frameworks And Considerations 8](#_Toc184118097)

[DATA SCIENCE PLatforms 10](#_Toc184118098)

**List of Tables**

[Table 1: Ways to Run Jupyter Notebooks 1](#_Toc176601036)

[Table 2: Important Python Data Science References 1](#_Toc176601037)

[Table 3: Common and Important Python Data Science Packages and Libraries 3](#_Toc176601038)

[Table 4: Sites with Sample Datasets and Data Catalogues 5](#_Toc176601039)

[Table 5: Sites with Visualization Examples References 5](#_Toc176601040)

[Table 6: Data Science News, Blogs and Articles 6](#_Toc176601041)

[Table 7: Meetups 7](#_Toc176601042)

[Table 8: AI Standards, Frameworks and Ethics Considerations 8](#_Toc176601043)

[Table 9: Relevant Generative AI References 9](#_Toc176601044)

# Data 601 Course Main Tools and Important References

* Jupyter Notebook (<https://jupyter.org/>) is a web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience. Various ways to run the Jupyter Notebook environment are shown in
* Python Documentation (<https://docs.python.org/3/>)

Once Jupyter Notebook is installed it can be run in multiple ways as described in Table 1.

Table 1: Ways to Run Jupyter Notebooks

|  |  |
| --- | --- |
| **Method to run Jupyter Notebooks** | **Comments** |
| Windows Jupyter Notebook Launcher (Windows) | Launch the Windows Desktop application directly. |
| Anaconda Navigator Jupyter Notebook (Windows and MacOS) | Launch Jupyter Notebooks from the Anaconda Navigator application. Applicable to both MacOS and Windows |
| Anaconda Prompt (Windows) | Open Anaconda prompt and type “jupyter notebook” |
| Jupyter Lab | Anaconda Prompt and Typing “jupyter-lab”. Jupyter Lab is a IDE like application from Jupyter with more capabilities than Jupyter Notebooks. |
| Terminal (MacOS) | Open Terminal and type “jupyter notebook” |
| Integrated Development Environment (IDE) | IDE’s such as Visual Studio Code allow to use the Jupyter Notebook outside of the Web‑browser environment. It also allows seamless transition when working on different codes in the IDE as well as managing different files. |
| Google Colab Research (<https://colab.research.google.com/>) | May not be an option when using internal, non-public data. Check with your organization. |

Once you are part of the team, my recommended approach is to use the tools that team is using. You should work on integrating into your team, and this will be critical and will support better communication, collaboration, debugging and exchange of information. For example, if your team uses a IDE (e.g., Visual Studio) that is probably a good time investment to make as it will allow you to continue to learn using their expertise and long term integrate better into how your team does its work and if this is the case your organization is probably investing into IDE‑specific training resources.

Table 2: Important Python Data Science References

|  |  |  |
| --- | --- | --- |
| **Title** | **Description** | **Link** |
| Python Documentation | Python Documentation | <https://www.python.org/doc/> and  <https://docs.python.org/3/> |
| Anaconda Included Packages List | Anaconda Distribution List of Packages | <https://docs.anaconda.com/anaconda/packages/pkg-docs/> |

Python style guide: <https://peps.python.org/pep-0008/>

# Data Science and Python Packages and Libraries

This table has a list of common and important Python data science packages and libraries. It also includes the link to the libraries Python Package Index (<https://pypi.org/>) as well as a short description. The table includes those packages and libraries included within Anaconda distribution. Alternatively, Google Collaboratory can call each library. Many of these will be discussed and used in the Data601 class while others are for reference for future classes or work projects. The library categories are as follows:

* Code
* Data: data loading and processing
* Math: mathematical, statistics, and scientific calculations
* Graph: Data visualization, plotting, charts, and graphics
* Text: Text manipulation
* NLP: Natural language processing
* Web: Interactions with web pages and application programming interfaces
* ML: Machine learning
* DL: Deep learning
* Survival Analysis
* Vision: Vision and image recognition problems.

Note that many graphics in many advanced websites use a combination of tools from to process the data, create initial visualizations, and enhance the final visualization. Also note that libraries could have overlapping scope, capabilities, and features. For example, a linear regression model can be created with Numpy, scikit‑learn, or sciPy. Deciding which library to use will depend on user familiarity with the module, speed, limitation and other considerations.

Within data science there are many topics that we learn in Data 601 and students are encouraged to continue reading about. These topics include but not limited to:

* Data processes:
  + Collection
  + Data cleaning, preparation, wrangling, transformation, processing, pipeline
* Working with data
  + Various types of data (e.g., integers, floats, text, dates, categorical, time series)
  + Various types of data collections (e.g., lists, arrays, dictionaries, data frames)
  + Data augmentation, synthetic data, data pipelines, algorithm and data bias, data balance
* Exploratory data analysis, data visualization, and statistics
* Machine Learning (ML)
* Supervised ML (e.g., train, test, split, sampling, fitting, predicting)
  + Regression
  + Classification (e.g., binary, multi-class, multi-label)
* Unsupervised ML Clustering (e.g., kmeans, DBSCAN)
* Model performance evaluation, metrics and related topics:
  + Feature selection
  + Model underfitting and overfitting
  + Accuracy, precision, recall, F1 score, confusion matrix
* Natural language processing (e.g., text normalization, similarity ranking, text classification, text clustering)
* Dashboarding and user interface design
* Ethics (e.g., fairness, security, privacy, civil rights)
* Communication of results

Other topics not discussed within Data 601 but may be discussed in other classes and are worth researching and continue learning about:

* Model optimization and hyperparameter tunning
* Version control, virtual environments, environment freezing
* Code deployment (e.g., websites, cloud service, docker and containers)
* Dimensionality reduction (e.g., PCA, TSNA)
* Ensemble Learning: merging prediction from multiple models including but not limited to ensemble models, bagging, boosting, stacking)
* Deep learning (e.g., transformers)
* Reinforcement learning
* Other Natural Language Processing (NLP) approaches:
  + Latent Dirichlet Allocation (LDA)
  + Word embeddings text models
  + Named entity recognition
  + Text sentiment analysis
  + Text summarization methods and models
  + Generative AI and large language models
* Knowledge Graphs and Network Analysis

Table 3: Common and Important Python Data Science Packages and Libraries

| **Library/ Package Title** | **Included in Anaconda** | **Category** | **Description and main use** |
| --- | --- | --- | --- |
| [PIP](https://pypi.org/project/pip/) | Yes | Code | Pip is the package installer for Python. |
| [Pandas](https://pypi.org/project/pandas/) | Yes | Data | Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. |
| [Numpy](https://pypi.org/project/numpy/) | Yes | Math | Numpy is a powerful N-dimensional array object, sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, useful linear algebra, Fourier transform, random number capabilities, and much more. |
| [Matplotlib](https://pypi.org/project/matplotlib/) | Yes | Graph | Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. |
| [Plotly](https://pypi.org/project/plotly/) | Yes | Graph | Plotly.py is an interactive, open-source, and browser-based graphing library for Python. |
| [Seaborn](https://pypi.org/project/seaborn/) | Yes | Graph | Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics. |
| [Re](https://pypi.org/project/re101/) (Regex or Regular Expressions) | Yes | Text | This package pertains specifically to regular expressions embedded inside Python and compiled with Python's [`re`] (<https://docs.python.org/3/library/re.html>) module. |
| [OS](https://docs.python.org/3/library/os.html) | Yes | OS | This module provides a portable way of using operating system dependent functionality. |
| [Requests](https://pypi.org/project/requests/) | Yes | Web | Requests is a simple, yet elegant, HTTP library. |
| [BeautifulSoup4](https://pypi.org/project/beautifulsoup4/) | Yes | Web | Beautiful Soup is a library that makes it easy to scrape information from web pages. It sits atop an HTML or XML parser, providing Pythonic idioms for iterating, searching, and modifying the parse tree. |
| [Scrapy](https://pypi.org/project/Scrapy/) | Yes | Web | Scrapy is a fast high-level web crawling and web scraping framework, used to crawl websites and extract structured data from their pages. |
| [Selenium](https://pypi.org/project/selenium/) | No | Web | Python language bindings for Selenium WebDriver. The selenium package is used to automate web browser interaction from Python. Can be used for dynamic JavaScript based websites. |
| [Scikit-learn](https://pypi.org/project/scikit-learn/) | Yes | ML | Scikit-learn is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license. |
| [Scipy](https://pypi.org/project/scipy/) | Yes | Math | SciPy is open-source software for mathematics, science, and engineering. The SciPy library depends on NumPy, which provides convenient and fast N-dimensional array manipulation. |
| [Statsmodels](https://pypi.org/project/statsmodels/) | Yes | Math | Statsmodels is a Python package that provides a complement to scipy for statistical computations including descriptive statistics and estimation and inference for statistical models. |
| [spaCy](https://pypi.org/project/spacy/) | No | NLP | SpaCy is a library for advanced Natural Language Processing in Python and Cython. |
| [NLTK](https://pypi.org/project/nltk/) | Yes | NLP | The Natural Language Toolkit (NLTK) is a Python package for natural language processing. |
| [Gensim](https://pypi.org/project/gensim/) | Yes | NLP | Gensim is a Python library for topic modelling, document indexing and similarity retrieval with large corpora. |
| [SQLite](https://docs.python.org/3/library/sqlite3.html) | Yes | DB | SQLite is a C library that provides a lightweight disk-based database that doesn’t require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language. |
| [Openpyxl](https://pypi.org/project/openpyxl/) | Yes | Excel | Openpyxl is a Python library to read/write Excel 2010 xlsx/xlsm/xltx/xltm files. |
| [Networkx](https://pypi.org/project/networkx/) | Yes | Network | NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks (e.g., knowledge graphs). |
| [Ipywidgets](https://pypi.org/project/ipywidgets/) | Yes | Dashboard | Ipywidgets, also known as jupyter-widgets or simply widgets, are interactive HTML widgets for Jupyter notebooks and the IPython kernel. |
| [Panel](https://pypi.org/project/panel/) | Yes | Dashboard | Panel provides tools for easily composing widgets, plots, tables, and other viewable objects and controls into custom analysis tools, apps, and dashboards |
| [Flask](https://pypi.org/project/Flask/) | Yes | Web | Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. |
| [Django](https://pypi.org/project/Django/) | No | Web | Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. |
| [TensorFlow](https://pypi.org/project/tensorflow/) | No | DL | TensorFlow is an open-source software library for high performance numerical computation (<https://www.tensorflow.org/>) |
| TensorFlow [Keras](https://pypi.org/project/keras/) | No | DL | Keras is TensorFlow’s module for deep learning. |
| [Pytorch](https://pypi.org/project/torch/) | No | DL | PyTorch is a Python package that provides two high-level features: (1) Tensor computation (like NumPy) with strong GPU acceleration and (2) Deep neural networks built on a tape-based autograd system. |
| [lifelines](https://lifelines.readthedocs.io/en/latest/) | No | Survival Analysis | Lifelines is a complete survival analysis library, written in pure Python. It handles right, left and interval censored data and contains the most popular parametric, semi-parametric and non-parametric models. |
| [OpenCV](https://pypi.org/project/opencv-python/) | No | Vision | OpenCV-Python is a library of Python bindings designed to solve computer vision problems (e.g., image and object detection, video capturing, image processing, etc.). |
| [Ultralytics/](https://pypi.org/project/ultralytics/)  [YOLO](https://pypi.org/project/ultralytics/) | No | Vision | Ultralytics YOLO (You Only Look Once) is Python package designed to solve object detection and image detection problems. |

# Referencem, Sites, and Examples

This section provides various references with sample datasets, data visualization examples, data science news, blogs, and articles.

Table 4: Sites with Sample Datasets and Data Catalogues

| **Title** | **Description** | **Link** | **Comment** |
| --- | --- | --- | --- |
| Github | Used by many developers to share code and collaborate. | <https://github.com/> | Class code is hosted in this website. |
| U.S. Government’s Open Data | Data, tools, and resources to conduct research, develop web and mobile applications, design data visualizations, and more. | <https://data.gov/> | Real world datasets that in some cases may be extremely large. |
| Kaggle | Host data science competitions, datasets, Jupyter notebooks, etc. | <https://www.kaggle.com/> | Good source for datasets and Jupyter Notebooks. |
| Google Research Data | Google Dataset Search Engine | <https://datasetsearch.research.google.com/> | Search engine for datasets. May send you to data in other websites in this table. |
| World Bank Data | Data collections by the World Bank | <https://datacatalog.worldbank.org/> | Also has collections from various sources. |
| Maryland Government Data | MD Open Data Website. | <https://opendata.maryland.gov/> | Real world datasets from Maryland’s State Government |
| U.S. Federal Government Federal Register | The website published all records that U.S. Federal agencies make public. | <https://www.federalregister.gov/> | Broad range of topics. Good data for practicing and performing NLP. |
| API for the Federal Register | <https://www.federalregister.gov/developers/documentation/api/v1> | One of the few open API’s that do not require a key and provides great functionality. |

The table contains reference websites that include sample galleries, how to use various plotting libraries with examples and articles on visualizations.

Table 5: Sites with Visualization Examples References

|  |  |  |  |
| --- | --- | --- | --- |
| **Title** | **Description** | **Link** | **Comments** |
| MatPlotLib Example Gallery | Python Package | <https://matplotlib.org/stable/gallery/index.html> | Included in Anaconda. |
| Seaborn Example Gallery | Python Package | <https://seaborn.pydata.org/examples/index.html> | Included in Anaconda. |
| Plotly Example Gallery | Python Package | <https://plotly.com/python/> | Included in Anaconda. |
| D3.JS | Website Plotting Library | <https://d3-graph-gallery.com/> | Used to deploy dynamic website plots. |
| Five Thirty Eight | Website | <https://fivethirtyeight.com/> | Excellent Websites that provide great examples on the capabilities of dashboards and data analytic visualizations. |
| The Pudding | Website | <https://pudding.cool/> |
| Information is Beautiful | Website | <https://informationisbeautiful.net/> |
| Visual Capitalist | Website | <https://www.visualcapitalist.com/> |
| Flowing Data | Website | <https://flowingdata.com/> |

The table below contains various websites with news, blogs and articles related to data analytics, data science, artificial intelligence, machine learning, natural language processing among other related topics.

Table 6: Data Science News, Blogs and Articles

|  |  |  |
| --- | --- | --- |
| **Title** | **Description** | **Link** |
| 50 years of data science | Paper | <https://courses.csail.mit.edu/18.337/2015/docs/50YearsDataScience.pdf> |
| A Very Short History Of Data Science | Article | <https://www.forbes.com/sites/gilpress/2013/05/28/a-very-short-history-of-data-science/> |
| Medium | Subscription based website with limited amount of free articles per month. | <https://medium.com/> |
| KD Nuggets Website | Website | <https://www.kdnuggets.com/> |
| Y Combinator News | Website | <https://news.ycombinator.com/> |
| Hacker Noon Website | Website | <https://hackernoon.com/> |
| Reddit Data Science Blogs | Reddit Website Blogs | <https://www.reddit.com/r/datascience/> |
| Data Elixir Newsletter | Website | <https://dataelixir.com/newsletters/> |
| Inside Big Data Website | Website | <https://insidebigdata.com/> |
| AI Google Blog | Blog | <https://ai.googleblog.com/> |
| Machine Learning Mastery Website | Website | <https://machinelearningmastery.com/> |

Table 7: Meetups

|  |  |
| --- | --- |
| **Title** | **Link** |
| Data Science Meetup | <https://www.meetup.com/topics/data-science/> |
| Data Works Meetup | <https://www.meetup.com/DataWorks/> |
| Statistical Seminars Meetup | <https://www.meetup.com/Statistical-Seminars-DC/> |

# AI Standards, And Ethics Frameworks And Considerations

The following table contains various references, standards and frameworks related to ethics

considerations.

Table 8: AI Standards, Frameworks and Ethics Considerations

|  |  |  |
| --- | --- | --- |
| **Organization** | **Description** | **Link** |
| National Institute of Standards and Technology (NIST) | NIST artificial intelligence references. | <https://www.nist.gov/artificial-intelligence> |
| NIST | NIST artificial intelligence (AI) risk management framework to better manage risks to individuals, organizations, and society associated with AI. | <https://www.nist.gov/itl/ai-risk-management-framework> |
| U.S. Government Accountability Office (GAO) | GAO AI framework report identifies key accountability practices centered around the principles of governance, data, performance, and monitoring to help federal agencies and others use AI responsibly. | <https://www.gao.gov/products/gao-21-519sp> |
| Institute of Electrical and Electromechanical Engineers (IEEE) | IEEE AI Standard | <https://standards.ieee.org/initiatives/artificial-intelligence-systems/> |
| IEEE 7000™-2021 | IEEE 7000™-2021 integrates ethical and functional requirements to mitigate risk and increase innovation in systems engineering product design. | <https://standards.ieee.org/news/2021/ieee-7000/> |
| Congressional Research Service (CRS); Generative AI and Data Privacy: A Primer | The CRS generative AI primer provides introduction to the concepts within generative AI. | <https://crsreports.congress.gov/product/pdf/R/R47569> |
| CRS; Generative AI and Copyright Law | The CRS generative AI and copyright law provides an  introduction to the concepts copyright law and AI. | <https://crsreports.congress.gov/product/pdf/LSB/LSB10922> |

Table 9: Generative AI References

|  |  |  |
| --- | --- | --- |
| **Title** | **Links** | **Description** |
| Gartner: Generative AI: What Is It, Tools, Models, Applications and Use Cases | <https://www.gartner.com/en/topics/generative-ai> | General discussion on Generative-AI. |
| Gartner: What Generative AI Means for Business | <https://www.gartner.com/en/insights/generative-ai-for-business> | General discussion on Generative-AI from a business perspective. |
| PricewaterhouseCoopers (PwC); Managing Generative AI Risks | <https://www.pwc.com/us/en/tech-effect/ai-analytics/managing-generative-ai-risks.html> | General discussion on Generative-AI. |
| Microsoft Azure; Generative‑AI Documentation and Guide | <https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/prompt-engineering> | Microsoft Azure Services (e.g, OpenAI) documentation and guides which discusse generative‑AI related techniques (e.g., prompt engineering) and other examples using MS Azure OpenAI services. |
| U.S. Department of Energy | <https://www.energy.gov/cio/department-energy-generative-artificial-intelligence-reference-guide> | Reference Guide include considerations for users and developers when using Generative AI tools. |

# DATA SCIENCE PLatforms

In class we use Python and related data science libraries. We specifically utilize the Anaconda Distribution which contains a curated list of Python and important data science libraries. However, many commercial companies develop their platforms which in many cases are built on top of Python programming language or have proprietary user interfaces that ease dashboard and tool development as well as data analysis and deployment of AI/ML/NLP. The table below provides a list of example data analytics and data science platforms all with different costs, capabilities, limitations, learning curves, and certifications. Also note that there are many low code or no‑code development platforms not included in this list.

Table 10: Sample of Commercial Platforms and Related Links

|  |  |
| --- | --- |
| **Platform** | **List of Training or Certifications** |
| MS Power Apps/Power Bi/Azure | <https://learn.microsoft.com/en-us/credentials/browse/> |
| Amazon Web Services (AWS) | <https://d1.awsstatic.com/training-and-certification/docs/AWS_certification_paths.pdf?pp=cert&c=exam&z=6>  <https://aws.amazon.com/certification/> |
| NVIDIA Deep Learning Institute (DLI) | <https://www.nvidia.com/en-us/training/> |
| NVIDIA GTC Conference | <https://www.nvidia.com/gtc/> |
| Google Cloud Services Platform | <https://grow.google/certificates/en_us/> |
| MicroStrategy | <https://www.microstrategy.com/education> |
| Tableau | <https://www.tableau.com/learn> |
| Hadoop | <https://hadoop.apache.org/> |
| Palantir Foundry | <https://www.palantir.com/platforms/aip/bootcamp/> |
| Sky Foudnry SkySpark | <https://skyfoundry.com/> |
| PNNL Inspire | <https://in-spire.pnnl.gov/> |