A beginner's guide to Data Science with Python

Make your way into the world of data with Python

Aditya Kousik Cotra Follow
May 5 · 4 min read ★

Data Science is one of the most sought fields in this century. Ever wonder why data is so important? Why researchers, scientists and business managers around the world are glued to their screens analyzing and manipulating data, finding better and efficient ways to use it? Because

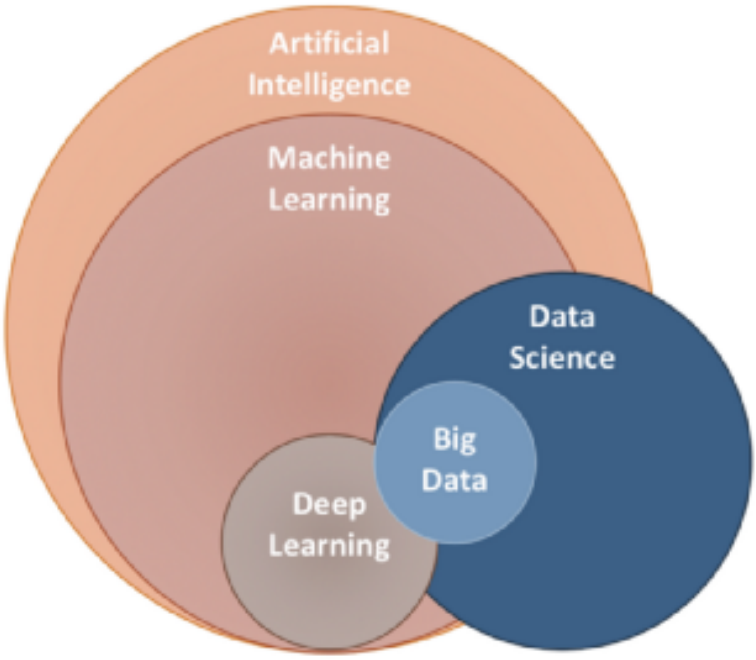
> Data is like fuel.

We humans always try to make our lives efficient and easy. But throughout the history of mankind, life has never been easy (that's a different issue). We are trying to make machines work for us, machines which can utilize information far more efficiently than human beings. So, how can you make a machine perform a particular task? We train the machine. And for that, we will need data. Sometimes lots of it.

There's a long way to go before machines can perform all the tasks like a human being. Let's dive into the basic areas in the field of data.

Know what you are dealing with

The science of data deals with a large variety of data. Data includes texts, numbers, images, audio and what not. It's not just one field, rather a combination of multiple fields focusing on analyzing and manipulating data in different ways for different purposes. Machine Learning, Deep Learning, Artificial Intelligence, Big Data and Data science are very popular today. It is necessary to know the fields of study and how they overlap.



Data science fields of study

- Data science: This field specifically deals with data acquisition, cleaning, visualisation and making decisions based on data. Additionally it also deals with building machine learning models and big data.
- Artificial Intelligence: This field focuses on building intelligent machines capable of performing tasks like humans.
- Machine learning: Building models to learn the data and predict outcomes with or without supervision is the primary focus of this field.
- Deep learning: This is the deeper version of machine learning as the name suggests. Creating multi-layered neural networks where advanced analysis is needed, learning data in depth where traditional machine learning could not reach.
- Big data: Using different techniques to deal with large amounts of data.

Important Libraries

Below are the important and basic libraries in python used for Data Science.

1. **Pandas** is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. Install pandas using:

```
pip3 install pandas
```

You can read files, manipulate cells, columns and a lot more.

```
import pandas as pd

>>> data = pd.read_csv('filename.csv') #this is a dataframe. you can
also use read_excel depending on the extension of your data file.
>>> data.head(n) #displays first n rows of the dataframe. Default 5
>>> data.columns #displays column names
>>> data.dtypes #displays data types for each column
>>> data.summary() #displays statistics for numeric columns
>>> data.dropna() #drops null rows or columns
>>> data.fillna(value) #fills null values with 'value'
>>> data.to_csv('transformed_data.csv') #saving dataframe to csv
file.
```

2. **Numpy** is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Install using:

```
pip3 install numpy
```

Basic operations:

```
#import the library
import numpy as np

>>> np.zeros((2, 3)) #array with 0's of specified shape
array([[ 0., 0., 0.], [ 0., 0., 0.]])
>>> np.arange(2, 3, 0.1) #values between 2,3 with diff 0.1
array([ 2. , 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9])
>>> np.linspace(1,4,6) #six uniformly spaced values between 1 and 4
array([ 1. ,  1.6,  2.2,  2.8,  3.4,  4. ])
```

These arrays are different from the list data types.

3. **Matplotlib** is a comprehensive library for creating static, animated, and interactive visualizations in Python.

Install using:

```
pip3 install matplotlib
```

Basic operations:

```
import matplotlib.pyplot as plt

plt.plot(x,y) #x and y must be array of values of same length
plt.scatter(x,y) #scatter plot
plt.show() #displaying plot
```

4. **Seaborn** is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

```
pip3 install seaborn
```

Basic operations:

```
import seaborn as sns

sns.scatterplot(x=column1, y=column2, data=data) #scatter plot
sns.lineplot(x=column1, y=column2, data=data) #line plot
```

5. **Scikit-learn** is a free software machine learning library for the Python with various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Install using:

```
pip3 install scikit-learn
```

Basic operations:

```
from sklearn.linear_model import LinearRegression

lr = LinearRegression(param1=value1) #initialize with app. params
lr.fit(X,y) #train the model
new_y = lr.predict(new_X) #predict new data
```

You can perform all kinds of statistical model training and evaluation using different built-in metrics, cross-validation methods and many other optimizing techniques.

Scikit-learn also **provides datasets for use**. See this.

Where to code?

I can recommend only two frameworks best suitable for coding for data science with python.

1. **Jupyter Notebook:** The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. Further, these notebooks can also be uploaded to GitHub.

You can install Jupyter notebook using:

```
pip3 install jupyter notebook
```

And then run it using:

```
jupyter notebook
```

The terminal runs the command and redirects you to *localhost* in your default browser.

2. **Google Colab:** Colab notebooks are Jupyter notebooks that are hosted by Colab and they execute code on Google's cloud servers, meaning you can leverage the power of Google hardware, including GPUs and TPUs, regardless of the power of your machine. All you need is a browser. Just go to this.

Colab is used extensively in the machine learning community with applications including:

- Getting started with TensorFlow
- Developing and training neural networks
- Experimenting with TPUs
- Disseminating AI research
- Creating tutorials

To get started with colab, see this.

Beginner Data Science Artificial Intelligence Data Visualization Machine Learning