What is Data Science?

- It is a branch of computer science where we study how to store, use, and analyze data for deriving information from it.
- Data Science is about data gathering, analysis, and decision-making.
- Data Science is about finding patterns in data, through analysis, and make future predictions.
- Data science is essentially a method of viewing and analyzing statistics and probability.

Core competencies of a data scientist

- The Data Scientist requires knowledge of a vast range of skills to perform the required tasks.
- Most of the times data scientists work in a team to provide the best results,
  - For example, someone who is good at gathering data might team up with an analyst and some talented in presenting the information.
  - It's a very difficult task to find a single person with all the required skills.
- Below are the areas in which a data scientist could find opportunity
  - Data Capture :
    - It is a process of managing data sources like databases, excel, pdf and text, etc.
    - It converts the unstructured data to structured data.
  - Analysis :
    - It requires knowledge of basic statistical tools.
    - It includes knowledge of specialized math tricks and algorithms for the analysis of data.
  - Presentations :
    - The presentation provides a graphical representation of the pattern.
    - It helps to represent the results of the data analysis to the end-user.

Creating the Data Science Pipeline

- Data science is partly art and partly engineering.
- The Data science pipeline requires the data scientist to follow particular steps in the preparation, analysis, and presentation of the data.
- General steps in the pipeline are
  - Preparing the data :
    - The data we gathered from various sources may not come directly in the structured format.

- We need to transform the data into a structured format.
- Transformation may require changing data types, the order in which data appears, and even the creation of missing data.

- Performing data analysis :
  - Data science provides access to a larger set of statistical methods and algorithms.
  - Sometimes a single approach may not provide the desired output, we need to use multiple algorithms to get the result.
  - The use of trial and error is part of the data science art.
- Learning from data :
  - As we iterate through various statistical analysis methods and apply algorithms to detect patterns, we begin learning from the data.
  - After learning from the data, the result of the algorithm may be different than initially, we predict the output.
- Visualizing :
  - Visualization means seeing the patterns in the data and then being able to react to those patterns.
  - It also means being able to see when data is not part of the pattern.
- Obtaining insights and data products :
  - The insights you obtain from manipulating and analyzing the data help you to perform real-world tasks. For example, you can use the results of an analysis to make a business decision.

Why Python?

- Python is the vision of a single person, Guido van Rossum, Guido started the language in December 1989 as a replacement for the ABC language.
- Grasping Python's core philosophy :
  - Guido started Python as a skunkworks project.
  - The core concept was to create Python as quickly as possible, yet created a flexible language, runs on any platform, and provides significant potential for extension.
  - Python is used to create an application of all types. It support four programming styles(programming paradigms)
  - Functional :
    - Treats every statement as a mathematical equation and avoids any form of state or mutable data.
    - The main advantage of this approach is having no side effects to consider.

- This coding style lends itself better than the others to parallel processing because there is no state to consider.
- Many developers prefer this coding style for recursion and lambda calculus.

- Imperative :
  - Performs computations as a direct change to program state.
  - This style is especially useful when manipulating data structures and produces elegant but simple code.
- Object-oriented :
  - Relies on data fields that are treated as objects and manipulated only through prescribed methods.
  - Python doesn't fully support this coding form because it can't implement features such as data hiding.
  - This is a useful coding style for complex applications because it supports encapsulation and polymorphism.
- Procedural :
  - Treats tasks as step-by-step iterations where common tasks are placed in functions that are called as needed.

Understanding Python's Role in Data Science

- Python has a unique attribute and is easy to use when it comes to quantitative and analytical computing.
- Python is widely used in data science and is a favorite tool along with being a flexible and open-sourced language.
- Its massive libraries are used for data manipulation and are very easy to learn even for a beginner data analyst.
- Apart from being an independent platform it also easily integrates with any existing infrastructure which can be used to solve the most complex problems.
- Python is preferred over other data science tools because of following features,
  - Powerful and Easy to use
  - Open Source
  - Choice of Libraries
  - Flexibility
  - Visualization and Graphics
  - Well supported

Considering Speed of Execution

- Analysis of data requires processing power.
- The dataset is so large that it may slow down a very powerful system.
- Following factors control the speed of execution for data science application
  - Dataset Size :
    - Data science relies on huge datasets in many cases.
    - The application type determines the size of the dataset in part, but dataset size also relies on the size of the source data.
    - Underestimating the effect of dataset size is deadly in data science applications, especially those that need to operate in real-time (such as self-driving cars).
  - Loading Technique :
    - The method we use to load data for analysis is critical, and we should always use the fastest.
    - Working with data in memory is always faster than working with data stored on disk.
    - Accessing local data is always faster than accessing it across a network.
    - Performing data science tasks that rely on network is probably the slowest method of all
  - Coding Style :
    - Anyone can create a slow application using any programming language by employing coding techniques that don't make the best use of programming language functionality.
    - To create fast data science applications, you must use best-of-method coding techniques.
  - Machine capabilities :
    - Running data science applications on a memory-constrained system with a slower processor is impossible.
    - The system you use needs to have the best hardware.
    - Given that data science, applications are both processor and disk-bound, you can't cut corners in any area and expect great results.
  - Analysis Algorithm
    - The algorithm you use determines the kind of result you obtain and controls execution speed.
    - We must experiment to find the best algorithm for a particular dataset.

Performing Rapid Prototyping and Experimentation

- Python is all about creating applications quickly and then experimenting with them to see how things work.
- The act of creating an application design in code without necessarily filling in all the details is called prototyping.
- Python uses less code than other languages to perform tasks, so prototyping goes faster.
- The fact that many of the actions you need to perform are already defined as part of libraries that you load into memory makes things go faster still.
- Data science doesn't rely on static solutions. You may have to try multiple solutions to find the particular solution that works best. This is where experimentation comes into play.
- After you create a prototype, you use it to experiment with various algorithms to determine which algorithm works best in a particular situation.

Using the Python Ecosystem for Data Science

- We need to load certain libraries to perform specific data science tasks in python.
- Following are the list of libraries which we are going to use in this chapter :
  - **NumPy** - Performing fundamental scientific computing.
  - **Pandas** - Performing data analysis using pandas
  - **Matplotlib** - Plotting the data
  - **SciPy** - Accessing scientific tools
  - **Scikit-learn** - Implementing machine learning
  - **Keras** and **TensorFlow -** Used for deep learning
  - **NetworkX -** Creating graphs
  - **Beautiful Soup -** Parsing HTML documents
- **NumPy :**
  - NumPy stands for *Numerical Python*
  - NumPy is a Python library used for working with arrays.
  - It provides a function to work with linear algebra, Fourier transform, and matrices.
  - It is an open-source project and we can use it freely.
  - It is a library consisting of multidimensional array objects and a collection of routines for processing those arrays.
  - Using NumPy, mathematical, and logical operations on arrays can be performed very easily.
  - In Python, we have lists that serve the purpose of arrays, but they are slow to process.

- NumPy provides an array object that is very much faster than traditional Python lists. the array object in NumPy is called ndarray.
- Arrays are very frequently used in data science, where speed and resources are very important.
- Features of NumPy are as follows.
  - powerful n-dimensional arrays
  - numerical computing tools
  - interoperable
  - performant
  - easy to use
  - open source

- **Pandas :**
  - 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.
  - It offers data structures and operations for manipulating numerical tables and time series.
  - The library is optimized to perform data science tasks especially fast and efficiently.
  - The basic principle behind pandas is to provide data analysis and modeling support for Python that is similar to other languages such as R.
- **Matplotlib :**
  - The matplotlib library gives a MATLAB like an interface for creating data presentations of the analysis.
  - The library is initially limited to 2-D output, but it still provides a means to express analysis graphically.
  - Without this library, we cannot create output that people outside the data science community could easily understand.
- **SciPy :**
  - The SciPy stack contains a host of other libraries that we can also download separately.
  - These libraries provide support for mathematics, science, and engineering.
  - When we obtain SciPy, we get a set of libraries designed to work together to create applications of various sorts, these libraries are
    - NumPy
    - Pandas
    - Matplotlib
    - Sympy

- **Scikit-learn :**
  - The Scikit-learn library is one of many Scikit libraries that build on the capabilities provided by NumPy and SciPy to allow Python developers to perform domain-specific tasks.
  - Scikit-learn library focuses on data mining and data analysis, it provides access to following sort of functionality:
    - Classification
    - Regression
    - Clustering
    - Dimensionality reduction
    - Model selection
    - Pre-processing
- **Keras and TensorFlow :**
  - Keras is an application programming interface (API) that is used to train deep learning models
  - An API often specifies a model for doing something, but it doesn't provide an implementation.
  - TensorFlow is an implementation for the keras, there are many other implementations for the keras like Microsoft's Cognitive Toolkit, CNKT and Theano
- **NetworkX :**
  - NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks (For example GPS setup to discover routes through city streets).
  - NetworkX also provides the means to output the resulting analysis in a form that humans understand.
  - The main advantage of using NetworkX is that nodes can be anything (including images) and edges can hold arbitrary data.
- **Beautiful Soup :**
  - Beautiful Soup is a Python package for parsing HTML and XML documents.
  - It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping.