Chapter 1 Introduction to Data Science

1.1 What is Data Science?

The term "**big data**" has become a hot topic in recent years due to the rapid growth in the size and scope of datasets in various sectors and advancement in technology. It refers to any collection of data sets so large or complex that is difficult to process using traditional data management techniques. The methods to analyze massive amounts of data and extract the information it contains is called **data science**. Both data science and big data evolved from statistics and traditional data management but are now considered to be distinct disciplines.

The characteristics of big data are often referred to as the four V's:

- **Volume** - How much data is there?
- **Variety** - How diverse are different types of data?
- **Velocity** - At what speed is new data generated?
- **Veracity** - How accurate is the data?

These four properties make big data different from the data found in traditional data management tools. Consequently, the challenges they bring can be felt in almost every aspect: data capture, curation, storage, search, sharing, transfer, and visualization. In addition, big data calls for specialized techniques to extract the insights.

Data science and big data are used almost everywhere in both commercial and non-commercial settings. Commercial companies in almost every industry use data science and big data to gain insights into their customers, processes, staff, completion, and products. Many companies use data science to offer customers a better user experience, as well as to cross-sell, up-sell, and personalize their offerings. Governmental organizations are also aware of data's value. Many governmental organizations not only rely on internal data scientists to discover valuable information, but also share their data with the public. You can use this data to gain insights or build data-driven applications.

1.2 Types of data

In data science and big data we will come across many different types of data. Each of them tends to require different tools and techniques. The main categories of data are:

- Structured
- Unstructured
- Natural language
- Machine-generated
- Graph-based
- Audio, video, and images
- Streaming

Structured data is data that depends on a data model and resides in a fixed field within a record. As such, it's often easy to store structured data in tables within databases or Excel files. SQL, or Structured Query Language, is the preferred way to manage and query data that resides in databases. Hierarchical data such as a family tree is also structured but it is hard to store it in a traditional relational database.

Unstructured data is data that is not easy to fit into a data model because the content is context-specific or varying. Examples of unstructured data include regular emails and posts on social media. Although email contains structured elements such as the sender, title, and body text, it is difficult to analyze the context of the email due to the variety in language. Similarly, it is complicated to analyse the context of a post on social media due to use of different symbols and emoticons.

Natural language is a special type of unstructured data. It poses some challenges to process because it requires knowledge of specific data science techniques and linguistics. The natural language processing community has had success in entity recognition, topic recognition, summarization, text completion, and sentiment analysis, but models trained in one domain do not generalize well to other domains.

Machine-generated data is information that's automatically created by a computer, process, application, or other machine without human intervention. Machine-generated data is becoming a major data resource and will continue to do so. The analysis of machine data relies on highly scalable tools, due to its high volume and speed. Examples of machine data are web server logs, call detail records, network event logs, and telemetry.

Graph-based or network data is data that focuses on the relationship or adjacency of objects. The graph structures use nodes, edges, and properties to represent and store graphical data. Graph-based data is a natural way to represent social networks, and its structure allows you to calculate specific metrics such as the influence of a person and the shortest path between two people. Examples of graph-based data can be found on many social media websites such as Twitter and LinkedIn.

Audio, image, and video are data types that pose specific challenges to a data scientist. Tasks that are trivial for humans, such as recognizing objects in pictures, turn out to be challenging for computers. For example, a company called DeepMind succeeded at creating an algorithm that is capable of learning how to play video games. This algorithm takes the video screen as input and learns to interpret everything via a complex process of deep learning.

Streaming data can take almost any of the previous forms. In addition, the data flows into the system when an event happens instead of being loaded into a data store in a batch. Although it is not quite a different type of data, streaming data is treated here as such because you need to adapt your process to deal with this type of information. Examples are live sporting or music events, and the stock market.

As an introductory course, we will focus on structured data which is more convenient for data analyzing and deploying algorithms in artificial intelligence. However, it would also be important to recognize other types of data for your future career in different disciplines.

1.3 Data science process

The data science process typically consists of six steps:

1. Setting the research goal
2. Data collection
3. Data preparation
4. Data exploration
5. Data modelling
6. Presentation and automation

(1) Setting the research goal

Data science is mostly applied in the context of an organization. When you are asked to perform a data science project, you will first prepare a project charter. This charter contains information such as what you are going to research, how the organization benefits from that, what data and resources you need, a timetable, and deliverables.

(2) Data collection

The second step is to collect data. You've stated in the project charter which data you need and where you can find it. In this step, you ensure that you can use the data in your program, which means checking the existence, quality of, and access to the data. Data can also be delivered by third-party companies and takes many forms.

(3) Data preparation

Data collection is an error-prone process; in this phase you enhance the quality of the data and prepare it for use in subsequent steps. This phase consists of three sub-phases: (i) data cleansing removes false values from a data source and inconsistencies across data sources; (ii) data integration enriches data sources by combining information from multiple data sources; and (iii)data transformation ensures that the data is in a suitable format for use in your models.

(4) Data exploration

Data exploration is concerned with building a deeper understanding of your data. You try to understand how variables interact with each other, the distribution of the data, and whether there are outliers. To achieve this you mainly use descriptive statistics, visual techniques, and simple modelling.

(5) Data modelling

In this phase you use models, domain knowledge, and insights about the data you found in the previous steps to answer the research question. You select a technique from the fields of statistics, machine learning, operations research, and so on. Building a model is an iterative process that involves selecting the variables for the model, executing the model, and model diagnostics.

(6) Presentation and automation

Finally, you present the results. These results can take many forms, ranging from presentations to research reports. Sometimes you'll need to automate the execution of the process because the business will want to use the insights you gained in another project or enable an operational process to use the outcome from your model.

1.4 Application

After understanding the basic concepts of data science, we will look into some application of data analytics and artificial intelligence which make use of various types of data.

(1) Map and traffic

Travelling to a new destination does not require much thought any longer. Rather than relying on confusing address directions, we can now easily open our phone's map app and type in our destination. So how does the app know about the appropriate directions, best way, and even the presence of roadblocks and traffic jams? A few years ago, only GPS (satellite-based navigation) was used as a navigation guide. However, artificial intelligence (AI) now provides users with a much better experience in their unique surroundings.

The app algorithm uses machine learning to recall the building's edges that are supplied into the system after the person has manually acknowledged them. This enables the map to provide simple visuals of buildings. Another feature is identifying and understanding handwritten house numbers, which assists travellers in finding the exact house they need. Their outline or handwritten label can also recognize locations that lack formal street signs.

The application has been trained to recognize and understand traffic. As a result, it suggests the best way to avoid traffic congestion and bottlenecks. The AI-based algorithm also informs users about the precise distance and time it will take them to arrive at their destination. It has been trained to calculate this based on the traffic situations. Several ride-hailing applications have emerged as a result of the use of similar AI technology. So, whenever you need to book a cab via an app by putting your location on a map, this is how it works.

(2) Face detection and recognition

Utilizing face ID for unlocking our phones and using virtual filters on our faces while taking pictures are two uses of AI that are presently essential for our day-by-day lives. Face recognition is used in the former, which means that every human face can be recognized. Face recognition is used in the above, which recognizes a particular face.

Intelligent machines often match human potential. Human babies begin to identifying facial features such as eyes, lips, nose, and face shapes. A face, though, is more than just that. A number of characteristics distinguish human faces. Smart machines are trained in order to recognize facial coordinates (x, y, w, and h; which form a square around the face as an area of interest), landmarks (nose, eyes, etc.), and alignment (geometric structures). This improves the human ability to identify faces by several factors. Face recognition is also used by government facilities or at the airport for monitoring, and security.

(3) Text and Language

When typing a document, there are inbuilt or downloadable auto-correcting tools for editors of spelling errors, readability, mistakes, and plagiarism based on their difficulty level. It should have taken a long time for us to master our language and become fluent in it. Artificially intelligent algorithms often used deep learning, machine learning, and natural language in order to detect inappropriate language use and recommend improvements. Linguists and computer scientists collaborate in teaching machines grammar in the same way that we learned it in school. Machines are fed large volumes of high-quality data that has been structured in a way that machines can understand. Thus, when we misspell a single comma, the editor will highlight it in red and offer suggestions.

For example, launched in 2009, Grammarly is a cloud-based typing assistant that reviews spelling, grammar, punctuation, clarity, engagement, and delivery mistakes. It uses artificial intelligence to identify and search for an appropriate replacement for the error it locates. It also allows users to customize their style, tone, and context-specific language. It is available as a downloaded program for use with desktop applications, as a browser extension, and as a smartphone keyboard.

(4) Healthcare

Infervision is using artificial intelligence and deep learning to save lives. In China, where there are insufficient radiologists to keep up with the demand for checking 1.4 billion CT scans each year for early symptoms of lung cancer. Radiologists essential to review many scans every day, which isn't just dreary, yet human weariness can prompt errors. Infervision trained and instructed algorithms to expand the work of radiologists in order to permit them to diagnose cancer more proficiently and correctly.

(5) Banking and finance

The banking and finance industry has a major impact on our daily lives which means the world runs on liquidity, and banks are the gatekeepers who control the flow. Did you know that artificial intelligence is heavily used in the banking and finance industry for things such as customer service, investment, fraud protection, and so on? The automatic emails we get from banks if we make an ordinary transaction, are a simple example. That's AI keeping an eye on our account and trying to alert us regarding any potential fraud. AI is now being trained to examine vast samples of fraud data in order to identify patterns so that we can be alerted before it happens to us. If we run into a snag and contact our bank's customer service, we are probably speaking with an AI bot. Even the largest financial industry use AI to analyze data in order to find the best ways to invest capital in order to maximize returns while minimizing risk.

Not only that, but AI is set to play an even larger role in the industry, with major banks around the world investing billions of dollars in AI technology, and we will be able to see the results sooner rather than later.

1.5 Computational tools for data science

Currently many big data tools and frameworks exist. The big data ecosystem can be grouped into technologies that have similar goals and functionalities. Data scientists use many different technologies, but not all of them. In this course, we will focus on the **Python** programming language and the **Jupyter Notebook** for developing the programs.

A **computer program** is a collection of instructions executable by a computer to perform a specific task. **Computer programming** refers to the process of building a computer program. Python is a computer programming language with a standard set of numerical and data visualization tools that are used widely in commercial applications, scientific experiments, and open-source projects.

Jupyter Notebook is a web-based application for creating and sharing computational documents. It offers a modern and powerful web interface to Python. To install, you may:

Step (1): download Anaconda (choose Python 3 version)

https://docs.anaconda.com/anaconda/install/windows/

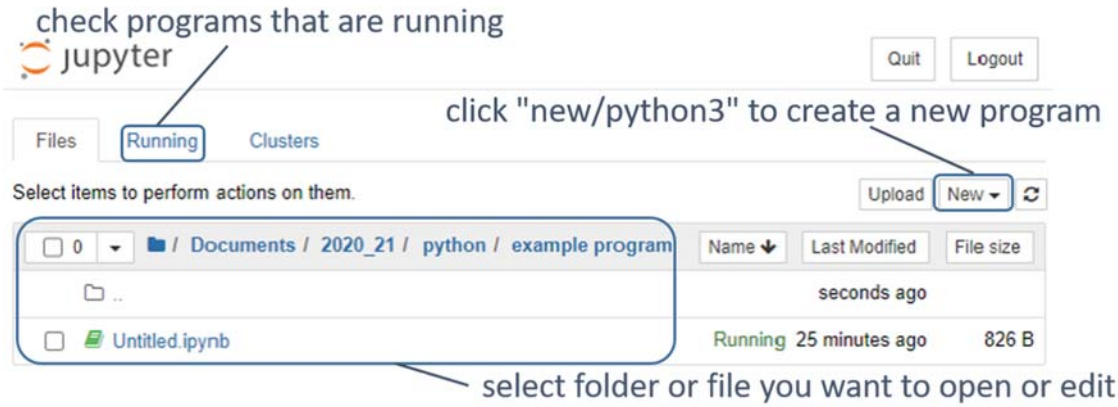
Step (2): install Anaconda

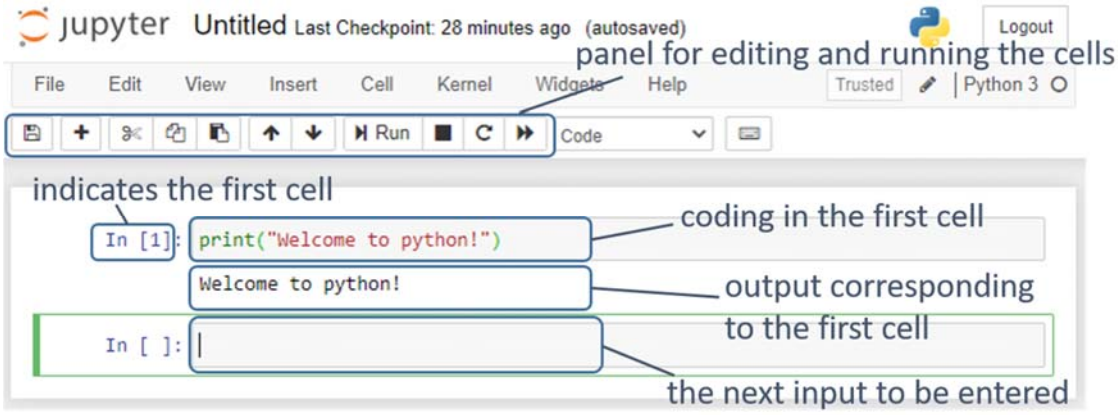
Step (3): to run Jupyter notebook, type: `jupyter notebook`

You may follow the guidelines on the website for more details.

https://test-jupyter.readthedocs.io/en/latest/install.html

A simple guideline of using Jupyter Notebook to implement python file can be referred to the following figures.





The advantage of using Jupyter Notebook is that it provides an interactive interface that allows user to view the outcome of the coding. Let's take a very simple example. One might want to evaluate the result of the arithmetic operation 1+1. Just type the coding on a cell and either click "Run" button or press Alt+Enter to execute the coding. The result:

```
In [1]: 1+1
Out[1]: 2
```

In Python, values can be stored as various data types such as:

- `int` - positive/negative/zero integers
- `float` - floating point real values
- `str` - strings which consist of multiple characters, enclosed by quotation marks
- `bool` - Boolean which is either True (1) / False (0), used in logical operations

On Jupyter Notebook, you may try to execute some simple calculation using the arithmetic operators as follows:

Operator	Name
+	addition
-	subtraction
*	multiplication
/	division
**	exponent
%	modulus
//	floor division
max(,)	maximum
min(,)	minimum

A **variable** is a reserved location to store values. To assign a value to a variable, use the syntax:

```
variable_name = value_assigned
```

For example, the coding below assigns the string 'ama' to x, the integer 123 to y, and the float number 1.23 to z. Click "Run" button or press Alt+Enter to execute the coding in a cell.

```
x = 'ama'
y = 123
z = 1.23
```

In Jupyter Notebook, you may check the value of each variable by calling it. You may also check their data type using `type()`.

```
x
```

```
'ama'
```

```
type(x)
```

```
str
```

```
y
```

```
123
```

```
type(y)
```

```
int
```

```
z
```

```
1.23
```

```
type(z)
```

```
float
```

1.6 Data structure

Data structure is a storage that is used to store and organize data. It is a way of arranging data on a computer so that it can be accessed and updated efficiently. To store data sequentially in the memory, we can use an array data structure.

In python, an array of numbers can be stored in various forms such as a **list**. A list is a collection of values which is ordered and changeable. To declare a list and assign values into it, we can list out the members separated by comma and enclosed in square brackets:

```
list_name = [member0, member1, member2, …]
```

Notice that each member is labelled by an **index** starting from 0 (instead of 1). To access a single member, we can use:

```
list_name[index]
```

To create a sliced list with consecutive members in the original list, we can use:

```
list_name[starting_index:stoping_index]
```

Notice that the term represented by the stopping index is not included in the sliced list.

For example we can create a list of numbers:

```
mylist = [2,3,5,7,11]
```

The five values are stored in a sequential order with index 0 to 4.

If we access the member with index 2 in `mylist`, it gives:

```
mylist[2]
```

```
5
```

If we form a sliced list from `mylist` with staring index 1 and stopping index 4:

```
mylist[1:4]
```

```
[3, 5, 7]
```

List is a primitive data structure in Python. In later sections, we will also use array from the Numpy library and Series from the Pandas library which have similar data structure as list but come with more powerful features.