Python for Machine Learning

Created	@Jan 18, 2021 1:37 PM
Created By	K Khanh Vương
Last Edited By	K Khanh Vương
Last Edited Time	@Jan 24, 2021 11:36 PM
Module	Introduction to Machine Learning
Status	
Type	Introduction to Machine Learning

Data types of Python

- There are 4 main data types in **Python**:
  - *int*
  - *float*
  - *string*
  - *booleans*
- However, **Sequences** can also be considered as data types
- To get the type name, use: `type()`

Typecasting in Python

- We must check the format of the variable before conduct typecasting. If the format does not meet the requirement, the we will get error
- For example, we can't conduct typecasting from `string` to `int` if the `string` has any non-number character

Relation Operations in Python

- There are two `booleans` values:
  - `True`
  - `False`
- Those two values must be written in uppercase. However, other values are written in lowercase, like `and`, `or`

Mathematics Operations in Python

- To get a `float`, use `/`
- To get an `int`, use `//`
- **Lambda** is an anonymous function: `x = lambda a: a + 3`

Sequences in Python

- There're four main sequences
  - **List**: `List = [A, B, C]`
  - **Set**: `Set = {A, B, C}`

    - **Tuple**: `Tuple = (A, B, C)`
    - **Dictionary**: `Dict = {"key_1": value_1, "key_2": value_2, "key_3": value_3 }`
- Only **List** is mutable
- **Set** will only has unique values
- **Keys** of a **Dict** must always be *unique* and *immutable*
- To get the length of a sequence, use `len()`

Ordered Sequences

- **Ordered Sequences** are those sequences that have index among it's values
- So that, `str` can also be considered as a sequence due to the fact that it has index
- To add a new item to the existed sequence, we can use `.append()` or `.extend()`
    - `.append()` will add only one item. If we passed multiple variables, then those will be zipped into a list
    - `.extend()` will add all of it's parameters to that sequence
- To create a new copy of the original sequence, we can use `A[:]`
- To delete an element, there are three different ways:
    - `del` will allow us to delete an object
    - `A.remove()` will allow us to delete a values in the sequence
    - `A.pop()` will allow us to pop a specific value at the index that passed to the method
- We can conduct indexing, slicing, even with negative index on those sequences:
    - **Indexing**: `A[0]`
    - **Slicing**: `A[start:end:step]`
    - **Concatenate**: `["A"] + ["B"] = ["A", "B"]`: `[A] + [B] = [A, B]`

- **Replicate**: `[A] * 3 == [A, A, A]`

Immutable Sequences

- Are those sequences that we cannot access it's values by index
- As a result, every methods that belong to this type of sequences cannot modify the original one, but must return a new copy of it

Operations on Dictionary

- **Add a new element**: `dict["new_key"] = new_value`
- **Delete an element:** `del(Dict["key"])`
- **Check the existence of an element**: `"key" in Dict`
- **Keys:** `Dict.keys()`
- **Value:** `Dict.values()`

Operation on Set

- **Add a new alement:** `set.add(new_element)`
- **Remove an element:** `set.remove(element)`
- **Verify if an element is in the set:** `element in set`
- **Verify if an element is in both sets:** `set_A & set_B` ⇒ This will return a new set that contains all of the element that both belong to `set_A` and `set_B`
- **Combine two sets:** `set_A.union(set_B)` ⇒ This will return a new set that contain both `set_A` and `set_B`
- **Verify if a set is a subset of another set:** `set_A.issubset(set_B)`

Iterate through sequences

- To get both `index` and `value` through a loop, we can use: `for index, value in enumerate(sequence)`
- We can also apply slicing in 2D array: `A[row_start:row_end, col_start:col_end]`

Sort a Sequence

- To sort a sequence, we can use:
  - `sort()` if we want to sort right on the original sequence
  - `sorted()` if we want to have a copy of the sequence that is sorted

Some other notes

- We can only create a `tuple` with only one element with this syntax: `tuple = (A,)`

Differences between List and Array in Python

- `list` is a type of linked-list, so that every element of a `list` must not be in the same type. Therefor, `list` is more flexible, and it's not fixed in size
- `array` is a sequence of memories, therefor, it can only contain elements with the same type. `array` must also be fixed in size, so it is much more efficient in math calculations because it computes faster than `list`

Reference

- https://medium.com/@paulrohan/python-list-vs-tuple-vs-dictionary-4a48655c7934

Function

- If we want to add an instruction to a function, we can think of it's document:
  - Create a document for a function with: `""" """`
  - Use `help()` to print out the document of that function
- To pass a sequence as a parameter to a function, we must use `*` at the beginning of the parameter's name: `def function(*parameter)`

Class

- `Class` has four main properties:
  - Abstract
  - Encapsulation
  - Inheritance
  - Polymorphism
- `Class` name must be uppercased the first letter
- `Class` contains all shared methods and properties, while `instance` of a `class` can contain private properties, but not private methods
- `Class`'s methods always have `self` as it's default parameter. `self` stands for the `instance` that call the method
- If we want to invoke a method in a subclass, we can use: `parent.__init__(self)`

Numpy

- We should use `numpy` because `numpy` is faster, and much more efficient in math calculation than traditional sequences
- `.shape()` in `numpy` will return a `tuple` that show the number of rows and columns

Pandas

- To create a new `dataframe` base on the existed one, we can use `df[['column']]`
- Note that:
  - `df['column']` will return a `series`
  - `df[['columns']]` will return a `dataframe`

File

- Best practice for open and the close a file is: `with open('path', 'mode') as file:`, this will automatically close the file after finish working on it
- We can check whether a file is closed with `.closed()`