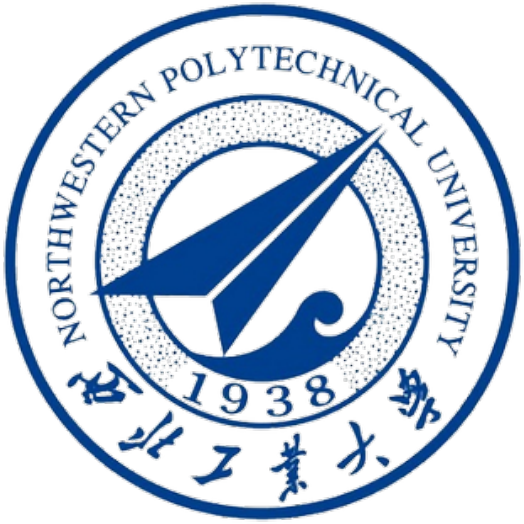

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Homework

Python data analysis and application

Submitted By:

ABID ALI

2019380141

abiduu354@gmail.com/ 3616990795@qq.com

Submitted To:

Professor 张维

zhangw@nwpu.edu.cn

Date of Submission: 24th April 2022

Contents

Answer to the question 1:

Six Comparison Operators in Python:

- Equal to (==)
- Not equal to (! =)
- Less than (<)
- Less than or equal to (<=)
- Greater than (>)
- Greater than or equal to (>=)

Answer to the question 2:

Similarities and Differences between 'Tuple' and 'List' in Python:

Similarities:

- They are data structures used to store data
- Index based access is possible

Differences:

- Lists are mutable, while tuples are immutable
- Syntax is different
- By default lists are of variable length, while tuples are of fixed length
- Python by default allocates comparatively less memory for tuples
- The implication of iterations is comparatively faster in case of tuples

Answer to the question 3:

In python we define a function with "def" keyword then we write the function identifier (name of the function) followed by parenthesis and a colon.

To write the function body we need to indent it with a "TAB" or 4 spaces, and put the write function in the body. In addition, a properly written function should have proper documentation explaining what the function does and what kind of parameters it has and what kind of input should be provided as parameters.

There are primarily 5 Types of Arguments in Python Function Definition:

- default arguments:

  ➔ Default arguments are values that are provided while defining functions.
  ➔ The assignment operator = is used to assign a default value to the argument.
  ➔ Default arguments become optional during the function calls.
  ➔ If we provide a value to the default arguments during function calls, it overrides the default value.
  ➔ The function can have any number of default arguments
  ➔ Default arguments should follow non-default arguments.

- keyword arguments:

  ➔ Functions can also be called using keyword arguments of the form kwarg=value.
  ➔ During a function call, values passed through arguments need not be in the order of parameters in the function definition. This can be achieved by keyword arguments. But all the keyword arguments should match the parameters in the function definition.

- positional arguments:

  ➔ During a function call, values passed through arguments should be in the order of parameters in the function definition. This is called positional arguments.

- arbitrary positional arguments:

  ➔ For arbitrary positional argument, an asterisk (*) is placed before a parameter in function definition which can hold non-keyword variable-length arguments. These arguments will be wrapped up in a tuple. Before the variable number of arguments, zero or more normal arguments may occur.

- arbitrary keyword arguments:

  ➔ For arbitrary positional argument, a double asterisk (**) is placed before a parameter in a function which can hold keyword variable-length arguments.

Answer to the question 4:

CODE:

```
# for regular expressions
import re
 # Make a regular expression
# for validating an Email
regex = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'
 # Define a function for
# for validating an Email
def check(email):
     # pass the regular expression
    # and the string into the fullmatch() method
    if(re.fullmatch(regex, email)):
        print("Valid Email")
    else:
        print("Invalid Email")
# Driver Code
if __name__ == '__main__':
    # Enter the email
    testerEmail = "johndoe2345@gmail.com"
    # calling run function
    check(email)
```

Answer to the question 5:

The four principles of object-oriented programming are encapsulation, abstraction, inheritance, and polymorphism.

Encapsulation

Encapsulation is the process of making certain attributes inaccessible to their clients and can only be accessed through certain methods. The inaccessible attributes are called private attributes, and the process of making certain attributes private is called information hiding. Private attributes begin with two underscores.

Abstraction

Python does not have a direct support for abstraction. However, abstraction is enabled by calling a magic method. If a method in a superclass is declared to be an abstract method, subclasses that inherit from the superclass must have their own versions of the said method. An abstract method in a superclass will never be invoked by its subclasses. But, the abstraction helps maintain a certain common structure in all of the subclasses.

Inheritance

Inheritance is considered the most important feature in an OOPS. Inheritance is the ability of a class to inherit methods and/or attributes of another class. The inheriting class is called the subclass or the child class. The class from which methods and/or attributes are inherited is called the superclass or the parent class.

Polymorphism

The word ‘polymorphism’ is derived from the Greek language, meaning ‘something that takes different forms’. Polymorphism is a subclass’s ability to customize a method as per need that is already present in its superclass. In other words, a subclass may either use a method in its superclass as such or modify it suitably whenever required.

Answer to the question 6:

Exceptions in Python

Python has many built-in exceptions that are raised when your program encounters an error (something in the program goes wrong).

When these exceptions occur, the Python interpreter stops the current process and passes it to the calling process until it is handled. If not handled, the program will crash.

For example, let us consider a program where we have a function A that calls function B, which in turn calls function C. If an exception occurs in function C but is not handled in C, the exception passes to B and then to A.

If never handled, an error message is displayed and our program comes to a sudden unexpected halt.

Raising Exceptions in Python

In Python programming, exceptions are raised when errors occur at runtime. We can also manually raise exceptions using the raise keyword.

Catching Exceptions in Python

In Python, exceptions can be handled using a try statement.

The critical operation which can raise an exception is placed inside the try clause. The code that handles the exceptions is written in the except clause.

A try clause can have any number of except clauses to handle different exceptions, however, only one will be executed in case an exception occurs.

In some situations, you might want to run a certain block of code if the code block inside try ran without any errors. For these cases, you can use the optional else keyword with the try statement.

The try statement in Python can have an optional finally clause. This clause is executed no matter what, and is generally used to release external resources.

Answer to the question 7:

Differences between 'Text' File and 'Binary' File:

Some of the differences include:

- ➔ In text file, text, character, numbers are stored one character per byte i.e., 32667 occupies 5 bytes even though it occupies 2 bytes in memory.
- ➔ In binary file data is stored in binary format and each data would occupy the same number of bytes on disks as it occupies in memory.
- ➔ In the text file, the newline character is converted to carriage-return/linefeed before being written to the disk.
- ➔ In binary file, conversion of newline to carriage-return and linefeed does not take place.
- ➔ Text files are used to store data more user friendly.

➔ Binary files are used to store data more compactly.

➔ In the text file, a special character whose ASCII value is 26 inserted after the last character to mark the end of file.

➔ In the binary file no such character is present. Files keep track of the end of the file from the number of characters present.

➔ Content written in text files is human readable.

➔ Content written in binary files is not human readable and looks like encrypted content.

Answer to the question 8:

Reading CSV Files in Python

A CSV (or Comma Separated Value) file is the most common type of file that a data scientist will ever work with. These files use a "," as a delimiter to separate the values and each row in a CSV file is a data record.

These are useful to transfer data from one application to another and is probably the reason why they are so commonplace in the world of data science.

The Pandas library makes it very easy to read CSV files using the read_csv() function.

Reading JSON Files in Python

JSON (JavaScript Object Notation) files are lightweight and human-readable to store and exchange data. It is easy for machines to parse and generate these files and are based on the JavaScript programming language. JSON files store data within {} similar to how a dictionary stores it in Python. But their major benefit is that they are language-independent, meaning they can be used with any programming language – be it Python, C or even Java!

Python provides a json module to read JSON files. You can read JSON files just like simple text files. However, the read function, in this case, is replaced by json.load() function that returns a JSON dictionary.

Reading Text from PDF File

Python package PyPDF can be used to achieve text extraction plus more. This package can also be used to generate, decrypting and merging PDF files

Reading text from word in python:

Several libraries exist that can be used to read and write MS Word files in Python. However, I will be using the python-docx module owing to its ease-of-use. Executing the following "pip" command in my jupyter cell to download the python-docx module

Extracting word:

To read a doc file, first I will import the docx module and then create an object of the Document class from the docx module. Then pass the path of the my_word_file.docx to the constructor of the Document class.

Answer to the question 9:

Supervised Learning:

Supervised learning is the most common subbranch of machine learning today. Typically, new machine learning practitioners will begin their journey with supervised learning algorithms.

Supervised machine learning algorithms are designed to learn by example. The name "supervised" learning originates from the idea that training this type of algorithm is like having a teacher supervise the whole process.

The most common example of classification is determining if an email is spam or not. With two classes to choose from (spam, or not spam), this problem is called a binary classification problem. The algorithm will be given training data with emails that are both spam and not spam. The model will find the features within the data that correlate to either class and create the mapping function mentioned earlier: Y=f(x). Then, when provided with an unseen email, the model will use this function to determine whether or not the email is spam.

Unsupervised Learning:

Unsupervised learning is a branch of machine learning that is used to find underlying patterns in data and is often used in exploratory data analysis. Unsupervised learning does not use labeled data like supervised learning, but instead focuses on the data's features. Labeled training data has a corresponding output for each input. When using unsupervised learning, I am not concerned with the targeted outputs because the goal of the algorithm is to find relationships within the data and group data points based on the input data alone. Supervised learning is concerned with labeled data in order to make predictions, but unsupervised learning is not.

The goal of unsupervised learning algorithms is to analyze data and find important features. Unsupervised learning will often find subgroups or hidden patterns within the dataset that a human observer may not pick up on. This is shown in the figure above. With the given image, you can probably pick out the subgroups, but with a more complex dataset, these subgroups may not be so easy to find. This is where unsupervised learning can help us.

Answer to the question 10:

K-Means Clustering is an unsupervised Learning Algorithm , which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

> It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.