ANLY 555: Data Science Python Toolbox
Deliverable #1: Design

Technical Resources:

- Commenting Standards:
  - https://www.python.org/dev/peps/pep-0257/#what-is-a-docstring
  - https://www.python.org/dev/peps/pep-0257/
- Documentation using Doxygen
  - http://www.doxygen.nl/
    - http://www.doxygen.nl/manual/
    - http://www.doxygen.nl/manual/docblocks.html#pythonblocks

Background

Throughout this course you will be designing and implementing a Data Science Toolbox using Python. There will be 4 (or 5) major deliverables and required, supporting discussion posts. The first deliverable will be focused on designing the class hierarchy, building the basic coding infrastructure, and beginning the documentation process.

Overview of Deliverables

1. Design
2. Implement DataSet Class and subclasses
3. Implement ClassifierAlgorithm Class, simpleKNNClassifier, and Experiment Class
4. Implement ROC and kdTreeKNNClassifier
5. ARM (greedy tree search ARM) OR hmmClassifier (dynamic programming) and final package

Software Design Requirements Overview

The toolbox will be implemented using OOP practices and will take advantage of inheritance and polymorphism. Specifically, the toolbox will consist of 3 main classes some of which have subclasses and member methods as noted below. You will also submit a demo script for each submission that tests the capabilities of your newly created toolbox.

1. Class Hierarchy
   a. DataSet
      i. TimeSeriesDataSet
      ii. TextDataSet
      iii. QuantDataSet
      iv. QualDataSet
   b. ClassifierAlgorithm

i. simplekNNClassifier
ii. kdTreeKNNClassifier
iii. hmmClassifier **
iv. graphkNNClassifier **

c. Experiment

** may be implemented in 5th deliverable for extra credit

2. Member Methods for each Super and Sub Class (subclasses will have more specified members as well to be added later). Each subclass will inherit superclass constructor. All other member methods will be overridden unless design deviation is well-justified.
    a. DataSet
        i. __init__(self, filename)
        ii. __readFromCSV(self, filename)
        iii. __load(self, filename)
        iv. clean(self)
        v. explore(self)
    b. ClassifierAlgorithm
        i. __init__(self)
        ii. train(self)
        iii. test(self)
    c. Experiment
        i. runCrossVal(self, k)
        ii. score(self)
        iii. __confusionMatrix(self)

Details for Deliverable #1. Design

There are 3 main components for your deliverable this week.

1. Using Python, you will implement the "framework" and "stubs" for the above classes, subclass, and member methods. In this context, "framework" refers to the class declarations, method declarations (along with parameter lists), and associated syntactic necessities, but otherwise the structures will be vacuous (there will be no substantial code that performs any task). The goal is simply to implement scaffolding / blueprints for the Toolbox. We will fill in the "substance" in future deliverables. The methods will be implemented as "stubs" meaning that they will also be largely vacuous with the exception of a print statement which simply prints a message confirming that the method was invoked. This is for testing purposes.
2. Using python you will implement a test script, test.py, that tests the functionality of your code. You will test to ensure that the following capabilities behave as desired: object instantiation, method invocation, inheritance, and polymorphism. You will test all constructors and methods. An example will be provided or discussed.
3. Using Doxygen (or another UML-like documentation tool), create documentation which illustratively describes the class hierarchy, member attributes, and member methods. The description should be VERY comprehensive and include structural and functional details. Take full advantage of DocString comments!

Academic Integrity

Refer to the guidelines specified in the *Academic Honesty* section of this course syllabus or contact me if you have any questions.

Include the following comments at the start of your source code files:

```
#* <FileName>.<file extension>
#*
#*  ANLY 555 <term year>
#*  Project <>
#*
#*  Due on: <Due Date>
#*  Author(s): <your name>
#*
#*
#*  In accordance with the class policies and Georgetown's
#*  Honor Code, I certify that, with the exception of the
#*  class resources and those items noted below, I have neither
#*  given nor received any assistance on this project other than
#*  the TAs, professor, textbook and teammates.
#*
```

These comments must appear **exactly** as shown above.

Submission Details

Upload (as instructed by your professor) a zip folder containing ALL files (`.py, .pdf, and/or .html files`). Use the following folder name: `<firstname><Lastname>P1. For example, I would create a folder named jeremyBoltonP2 which contained all files. I would then zip this folder creating file jeremyBoltonP1.zip . I would then submit this zip file.` Late submissions will be penalized heavily. If you are late you may turn in the project to receive feedback but the grade may be zero. In general, requests for extensions will not be considered.

Programming Skills

The programming skills required to complete this assignment include:

- Multiple File Organization
- UML Design
- OOP Design
- Inheritance
- Polymorphism

Grade Rubric (Knowing what constitutes each category is part of the assignment.)

Correct Design:	25
Correct Functionality:	25
Good Documentation:	25
Comprehensive Testing:	25
TOTAL:	100