**Task 2: Wordcount in Python**

Count the words included in file: file01\_Hd\_Sp\_Freq.txt and report the result.

**Task 3: Count Frequencies of Specific Words in Python**

Now use the same file: file01\_Hd\_Sp\_Freq.txt in order to compute the three words encountered with the highest frequencies.

**Task 4: Pi Estimation**

Spark can also be used for compute-intensive tasks. This code estimates π by "throwing darts" at a circle. We pick random points in the unit square ((0, 0) to (1,1)) and see how many fall in the unit circle. The fraction should be π / 4, so we use this to get our estimate. Implement and run the program --- you only need to configure the output. Show result.

**Task 5: Search Text and Create RDDs: Practice from SCALA to Python**

Convert the SCALA code below to Python, run it and show results. If there are errors in the given code that prevent you from obtaining the expected result, please make a simple fix and show result.

**from pyspark import SparkContext**

**# Insert your own code, transform the code below:**

**val lines = sc.textFile("file01\_Hd\_Sp\_Freq.txt ")**

**// transformed RDDs**

**val selfish = lines.filter(\_.startsWith("I"))**

**val messages = selfish.map(\_.split("\t")).map(r => r(1))**

**messages.cache()**

**// action 1**

**messages.filter(\_.contains("Spark")).count()**

**Task 6: Use PySpark to implement the Common Friends Problem.**

Please submit the code, the solution, and your output.

**Task 7: Use PySpark to implement the MapReduce problem that finds the 30 most frequent words beginning with each letter.**

Please submit the code, the solution, and your output.

Use the following document: <http://www.gutenberg.org/files/1342/1342-0.txt>