|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CST 2101 – Business Intelligence Programming**  **Project 2: Text Analysis** | | |  | **Due: 22 Dec 2017 – 11:59pm** | |
| **Background** | | |  | **Submission (15 marks)** | |
| Text analysis and the statistical distribution of words can tell us a lot about who wrote it and such an analysis of corporate emails was a key factor in the prosecution of Enron corporation executives for insider trading. Specifically, the increasingly frequent use of pronouns in email was cited as a metric for detecting deceptive communication in that case.    The client would eventually like to work along these lines, but for now wants you to expose some concepts about processing text so that they can determine how they to better analyze corporate documents as a first step.  Since their internal documents are sensitive they have asked you to analyze text copy of three novels: Charles Dickens’ A Tale of Two Cities, Leo Tolstoy’s War and Peace, and Victor Hugo’s Les Miserables.  Each of these files is sourced from the Gutenberg Project  ([www.gutenberg.org](http://www.gutenberg.org)). | | |  | You will submit the following files compressed into a 7-zip archive. The archive name will be formatted as follows:  LastName\_FirstName\_CST2101\_Project2.7z   1. All python files for your program implementation. 2. Preprocessing files (6). 3. Design and Test documents (5) 4. The output text files (3)   Submit it to Blackboard under the Project #2 Text Analysis. | |
| **Marking Guide** | | |  | Each criteria will be marked as follows to the maximum allowed by the category.   |  |  | | --- | --- | |  |  | | Missing or improper use affecting a minor element of the objective | -0.5 | | Missing or improper use affecting a major element of the objective | -1 | |  |  | | Run time or logic errors. | -1 | | |
| The assignment will be marked out of 15 using the following guide:   * The submission follows the project instructions. (1 mark) * The submission provides a functioning intuitive graphical user interface. (2 marks) * The submission demonstrates the correct use of classes/methods (must use at least two classes) (2 marks) * The submission demonstrates the correct use of loops and lists. (2 marks) * The program does not contain any logic or runtime errors. (2 marks) * Proper naming conventions for variables, classes etc. (1.0 marks) * The output text files are generated as per the specification. (1.0 mark) * Sufficient header and inline documentation. (1.0 mark) * GUI Design document reflects your GUI (1.0 mark) * Program Design document flow is easy to read and reflects your general logic (1.0) * Test plan covers basic functionality and exception handling (1.0 mark)   I will assess the submission by trying to run one valid and one invalid file through your program as well as a review of all submission documents. | | |  |
|  |
| Project Area | Requirement | | |
| General programming | * Must use Python 3, with meaningful naming and lower camel case style. * Must contain header documentation that describes the purpose of the program, the author and the date. * Must contain sufficient inline documentation for others to understand logic. * Must properly use classes, loops and lists in the program. | | |
| Preprocessing to Clean Data (using text editor – not in program) | * Must remove licensing terms * May remove table of contents * Must save removed text as **"<novel>\_removed.txt".** * Must save clean text as **"<novel>\_clean.txt".** | | |
| Design - GUI | * Must include identification of input and output widgets. * May be designed in Excel. * Must be named **Design\_GUI.<extension>** | | |
| Design - Program | * Must covey the general logic of the program including classes/methods. * May be constructed as “comments” in a .py file (no programming) * Must be named **Design\_Program.<extension>** | | |
| Development and Testing | * A small paragraph for development and testing purposes must be used. * The file must be named **Sample.txt** * Must create a test plan named **Test\_Plan.<extension>** * Must create a test output name **Sample\_Analysis.txt** | | |
| Input | * Must accept text files for input * May allow the user to choose the text file for processing. * Must allow the user to choose to start processing or exit without processing any text files | | |
| Processing | * Must be able to process a text file as input: * Must be able to count the:   + total # of words   + total # of occurrences for each word.   + total # of characters in the text document.   + total # of blank spaces in the text document. * Must be able to calculate the percentage of blank spaces as:   + total # of blank spaces divided by total # of characters, multiplied by 100. * Must implement exception handling in two areas:   + when opening the text file for reading and,   + when writing the output file. | | |
| Output | * Must create an output text file for the chosen text formatted to the client’s requirements named **"<novel>\_analysis.txt".** * Must advise the user when:   + processing has completed AND   + the name/location of the output file. * Must advise the user when:   + an exception has occurred AND   + the type of exception | | |
| Client Text Output Requirement | * Text analysis output must be clear and easy to read * Must include appropriate headers and data for:   + Name of text:   + Total Non-blank Character Count:   + Total Blank Character Count:   + Percentage Blank Character:   + Total Word Count:   + Word, Count: (each word and count is on a separate line) | | |