The objective of this deliverable is to enhance and correct the Refactored-BAS.jar ROC interface
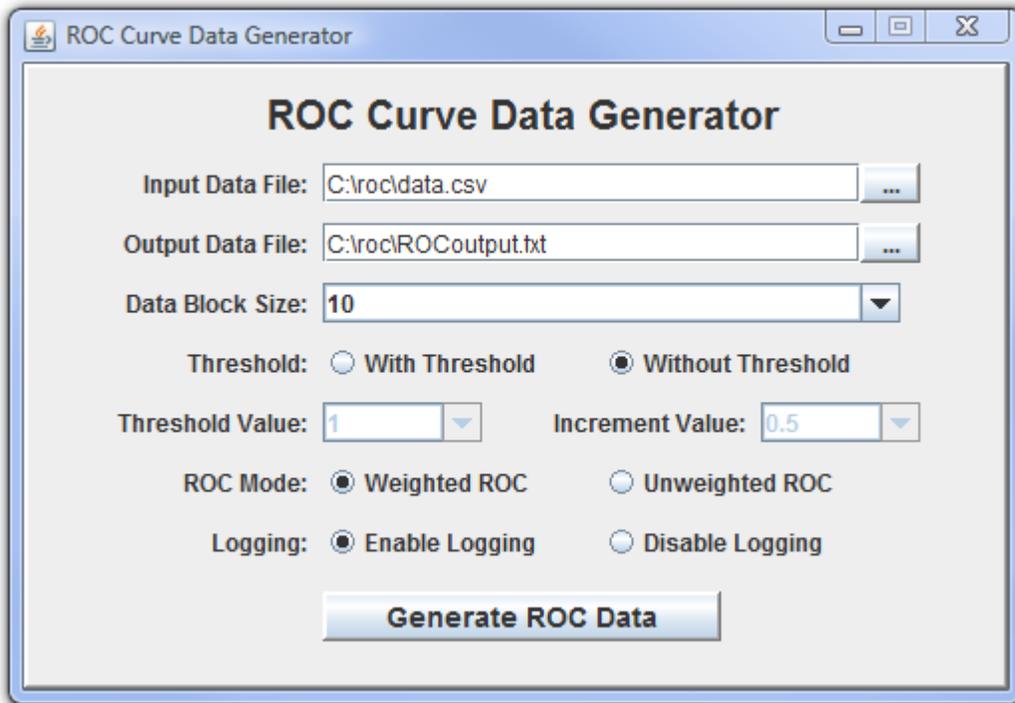
1. **Review the existing ROC Curve Generator**
   - These instructions assume that BAS is installed in the c:\it691 directory. Adjust the reference to match your installation.
   - Launch the existing ROC Curve Generator by running "roc.bat" from your BAS installation directory. The following screen should display:


2. Using Eclipse, setup and open the Java project for the ROC Curve Generator. Instructions have previously been provided and will be provided again upon request.

3. The Data Block Size must match the number of Nearest Neighbors selected. The current version supports block sizes of 10, 15, 20, 25, etc. The choices and underlying logic should be adjusted to reflect 9, 15, 21, and user defined. The formula for computing the loop range must change to reflect the block size. The formula for calculating the loop range or upper bound is:

$$\frac{k + (k + 1)}{2} + 1$$

For a Data Block Size for 9 Nearest Neighbors, the loop upper bound is:

$$\frac{9 + (9 + 1)}{2} + 1 = 46$$

- Another way to understand this for the block size to support a 9NN example is:

    9+8+7+6+5+4+3+2+1 = 45

    45 + 1 = 46

- The pick-list should change to reflect 9, 15, 21, and user defined value.
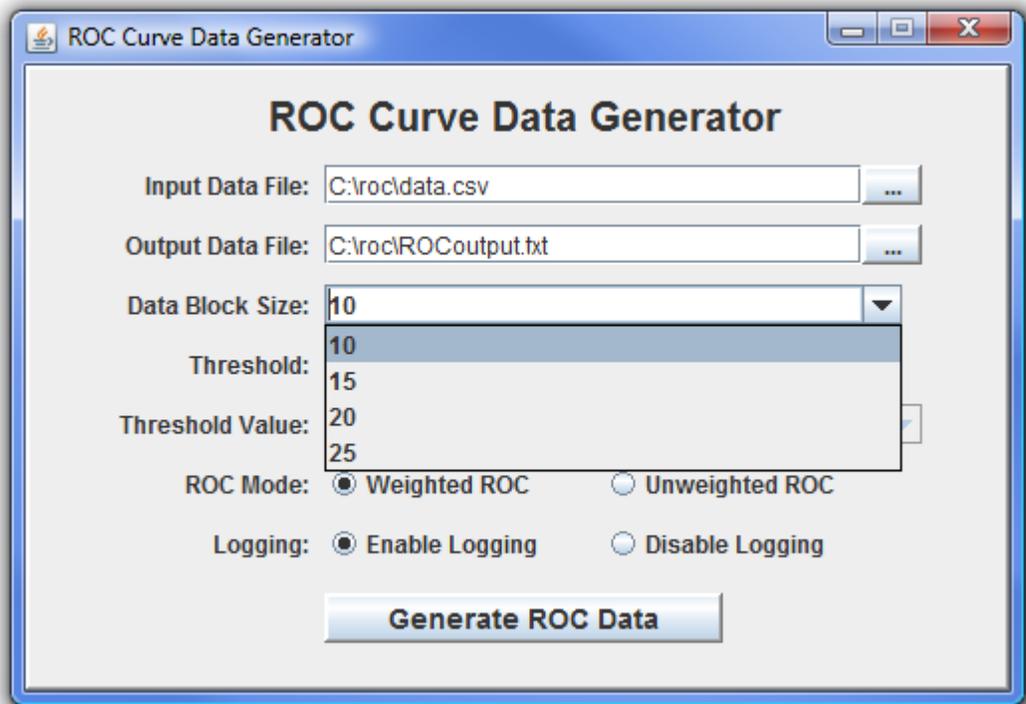
4. **Correct Threshold Code**

The current algorithm for calculating the threshold data is incorrect. The thresh.pl script corrects the program logic and produces the expected output. The objective of this step is to recode the Perl threshold script in Java and replace the incorrect threshold code in the Roc Curve Data Generator program.

The comments, (#) should explain the code. All of the debug statements in the Perl script can be removed from the Java code. To understand the program logic, you can turn on various debug statements. Simply comment out the $ debug = 0; line and un-comment the $debug = line at the beginning of the script. '$debug = 2 | 4 | 128' will show the output of the 2, 4 and 128 debug statements. Statements to trap output are tagged at various points in the script. You can select as few as no debug arguments, or all of them, separated by the '|' symbol.

The final Java code should produce results that include the threshold value, FAR, FRR, FAR%, FRR% in a text or comma delimited .csv format that is easy to import into Excel or other chart graphing software. Eventually, the charts will be created natively using Java graphing libraries.