**DV Program UML Description**

**Variable:**

|  |  |
| --- | --- |
| **Name:** | **Type:** |
| Description: | |

**Function:**

|  |  |
| --- | --- |
| Name: | Params: |
| Return type: | |
| Description: | |

**Java Files:**

**AngleSliders**

Creates a slider for a dataset feature.

**Functions/Methods**

|  |  |
| --- | --- |
| Name: createSliderPanel | Params: String fieldName, int angle, int index |
| Return type: void | |
| Description: Creates angle slider for a dataset feature. | |

**ColorOptionsMenu**

Creates a slider for a dataset feature.

**Functions/Methods**

|  |  |
| --- | --- |
| Name: ColorOptionsMenu | Params: Point mouseLocation |
| Return type: N/A | |
| Description: Constructor for ColorOptionsMenu. Creates a ColorOptionsMenu on mouseLocation. | |

**ConfusionMatrices**

Holds functions for setting up and creating graphs.

**Variables/Attributes**

|  |  |
| --- | --- |
| Name: upper | Type: ArrayList<double[]> |
| Description: holds overlapping endpoints of upper graph | |
| Name: lower | Type: ArrayList<double[]> |
| Description: holds overlapping endpoints of lower graph | |
| Name: LDAFunction | Type: ArrayList<Double> |
| Description: holds angles and threshold of for function created with data-without overlap | |

**Functions/Methods**

|  |  |
| --- | --- |
| Name: generateConfusionMatrices | Params: void |
| Return type: void | |
| Description: generates all data, data-without overlap, overlap, and worst case confusion matrices | |
| Name: getAllDataConfusionMatrix | Params: void |
| Return type: void | |
| Description: creates confusion matrix with all data within the subset of utilized data (data is separated optimally with Linear Discriminant Analysis) | |
| Name: getDataWithoutOverlapConfusionMatrix | Params: void |
| Return type: void | |
| Description: creates confusion matrix with all non-overlapping data within the subset of utilized data (data is separated optimally with Linear Discriminant Analysis) | |
| Name: getOverlapConfusionMatrix | Params: void |
| Return type: void | |
| Description: creates confusion matrix with overlapping data within the subset of utilized data (data is separated optimally with Linear Discriminant Analysis) | |
| Name: getWorstCaseConfusionMatrix | Params: void |
| Return type: void | |
| Description: creates confusion matrix with overlapped data within the subset of utilized data (data is separated with a function built with dataWithoutOverlap) | |
| Name: createCSVFileForConfusionMatrix | Params: ArrayList<ArrayList<double[]>> data |
| Return type: void | |
| Description: creates a csv file with given data | |
| Name: LDAForConfusionMatrices | Params: boolean storeFunction |
| Return type: ArrayList<String> | |
| Description: For a created csv file, uses linear discriminant analysis to create an optimal function, then generates a confusion matrix. Outputs angles and threshold if storeFunction is true. | |

**ConfusionMatrixMenu**

Creates a slider for a dataset feature.

**Functions/Methods**

|  |  |
| --- | --- |
| Name: ConfusionMatrixMenu | Params: Point mouseLocation |
| Return type: N/A | |
| Description: Constructor for ConfusionMatrixMenu. Creates a ConfusionMatrixMenu on mouseLocation. | |

**DataObject**

Holds data for a given class.

**Variables/Attributes**

|  |  |
| --- | --- |
| Name: className | Type: String |
| Description: name of class | |
| Name: data | Type: double[][] |
| Description: holds n-D datapoints for a given class | |
| Name: coordinates | Type: double[][] |
| Description: holds x and y locations for each dimension in each datapoint | |

**Functions/Methods**

|  |  |
| --- | --- |
| Name: DataObject | Params: String name, double[][] dataValues |
| Return type: N/A | |
| Description: Constructor for DataObject. Creates DataObject with given class name and dataValues. | |
| Name: updateCoordinates | Params: double[] angles |
| Return type: double | |
| Description: Generates the x and y coordinates for every dimension and datapoint in a given DataObject. Returns the scaling for a graph. | |
| Name: generateCoordinates | Params: double[] dataPoint, double[] angles |
| Return type: double[][] | |
| Description: returns the x and y coordinates for every dimension in a given datapoint | |
| Name: getXYPoint | Params: double value, double angle |
| Return type: double[] | |
| Description: returns x and y locations for a given value and angle | |

**DataSetup**

Takes in csv file of data and creates DataObjects for each class.

**Functions/Methods**

|  |  |
| --- | --- |
| Name: setupWithData | Params: File dataFile |
| Return type: | |
| Description: | |
| Name: setupImportData | Params: File importFile |
| Return type: | |
| Description: | |
| Name: getClasses | Params: String[][] stringData |
| Return type: | |
| Description: | |
| Name: checkClass | Params: String[] stringData |
| Return type: | |
| Description: | |
| Name: getStringFromCSV | Params: File dataFile |
| Return type: | |
| Description: | |
| Name: purgeID | Params: String[][] stringData |
| Return type: | |
| Description: | |
| Name: purgeClasses | Params: String[][] stringData |
| Return type: | |
| Description: | |
| Name: getFieldNames | Params: String[][] stringData |
| Return type: | |
| Description: | |
| Name: stringToNumerical | Params: String[][] stringData |
| Return type: | |
| Description: | |
| Name: normalizeData | Params: double[][] data |
| Return type: | |
| Description: | |
| Name: separateByClass | Params: double[][] data |
| Return type: | |
| Description: | |
| Name: createDataObjects | Params: ArrayList<double[][]> separateClasses |
| Return type: | |
| Description: | |
| Name: addImportData | Params: double[] data, int classIndex, boolean original |
| Return type: | |
| Description: | |
| Name: manualMinMaxEntry | Params: String message |
| Return type: | |
| Description: | |

**DataVisualization**

Holds functions for setting up and creating graphs.

**Functions/Methods**

|  |  |
| --- | --- |
| Name: optimizeSetup | Params: void |
| Return type: void | |
| Description: Uses linear discriminant analysis to find the best angles for a visualization. Then finds the optimal threshold, which class is lower (has lower endpoint positions), and the overlapping area of the visualization. | |
| Name: createCSVFile | Params: void |
| Return type: void | |
| Description: creates csv file for the linear discriminant analysis python program to use | |
| Name: LDA | Params: void |
| Return type: void | |
| Description: creates python process that runs linear discriminant on data within a created csv file | |
| Name: optimizeThreshold | Params: void |
| Return type: void | |
| Description: searches for the optimal threshold within a 60 tick range (15% of total domain) | |
| Name: optimizeVisualization | Params: void |
| Return type: void | |
| Description: Uses gradient search to find optimal angles within a 5 degree range. optimizeThreshold is also used on every set of generated angles. | |
| Name: undoOptimization | Params: void |
| Return type: void | |
| Description: reverses previous optimization attempt | |
| Name: getAccuracy | Params: void |
| Return type: void | |
| Description: gets current accuracy of angle and threshold setup of graph | |
| Name: getOverlap | Params: void |
| Return type: void | |
| Description: gets the overlap area of a visualization | |
| Name: drawGraphs | Params: void |
| Return type: void | |
| Description: creates upper and lower graphs | |
| Name: addGraph | Params: void |
| Return type: boolean | |
| Description: Creates graph. Returns true if graph is scaled. | |

**DV**

Main handler for UI. Creates main panel, menu bar, toolbar, sliders, and graphs.

**Variables/Attributes:**

|  |  |
| --- | --- |
| Name: domainSlider | Type: RangeSlider |
| Description: slider for the subset of utilized data for all classes | |
| Name: overlapSlider | Type: RangeSlider |
| Description: slider for the subset of overlapped data for all classes | |
| Name: thresholdSlider | Type: JSlider |
| Description: slider for the threshold location | |
| Name: angleSliderPanel | Type: JPanel |
| Description: panel for angle adjustment sliders |  |
| Name: confusionMatrixPanel | Type: JPanel |
| Description: panel for confusion matrices |  |
| Name: graphPanel | Type: JPanel |
| Description: panel for graphs | |
| Name: graphDomainPanel | Type: JPanel |
| Description: panel for graphPanel and sliders | |
| Name: allDataCM | Type: JTextArea |
| Description: confusion matrix for all data within subset of utilized data for all classes (data is optimally separated with Linear Discriminant Analysis) | |
| Name: dataWithoutOverlapCM | Type: JTextArea |
| Description: confusion matrix for non-overlapping data within subset of utilized data for all classes (data is optimally separated with Linear Discriminant Analysis) | |
| Name: overlapCM | Type: JTextArea |
| Description: confusion matrix for overlapping data within subset of utilized data for all classes (data is optimally separated with Linear Discriminant Analysis) | |
| Name: worstCaseCM | Type: JTextArea |
| Description: confusion matrix for overlapping data within subset of utilized data for all classes (data is separated with a function built with dataWithoutOverlap) | |
| Name: graphPane | Type: JScrollPane |
| Description: scrolling pane for graphs | |
| Name: anglesPane | Type: JScrollPane |
| Description: scrolling pane for angle sliders | |
| Name: confusionMatrixPane | Type: JScrollPane |
| Description: scrolling pane for confusion matrices | |
| Name: mainFrame | Type: JFrame |
| Description: frame for the whole DV Program | |
| Name: domainLines | Type: Color |
| Description: color for subset of utilized data marking lines | |
| Name: overlapLines | Type: Color |
| Description: color for the subset of overlapped data marking lines | |
| Name: thresholdLine | Type: Color |
| Description: color for the threshold marking line | |
| Name: graphColors | Type: Color[] |
| Description: colors for upper and lower graphs | |
| Name: showBars | Type: boolean |
| Description: whether to show or hide bar graph of endpoints for each graph | |
| Name: drawOverlap | Type: boolean |
| Description: whether to draw only the normal graphs or only the overlapping portions | |
| Name: domainActive | Type: boolean |
| Description: Whether the subset of utilized is active or not. If inactive, all data is used. | |
| Name: domainArea | Type: double[] |
| Description: locations for upper and lower extremes of the subset of utilized data | |
| Name: upperClass | Type: int |
| Description: current class being visualized on upper graph | |
| Name: lowerClasses | Type: ArrayList<Boolean> |
| Description: list of classes are being visualized on the lower graph | |
| Name: showPopup | Type: boolean |
| Description: whether to show a popup warning users about graph scaling | |
| Name: overlapArea | Type: double[] |
| Description: locations for upper and lower extremes of the subset of overlapped data | |
| Name: threshold | Type: double |
| Description: location of threshold | |
| Name: prevThreshold | Type: double |
| Description: previous threshold before optimizing | |
| Name: upperIsLower | Type: boolean |
| Description: whether the upper graph is lesser (has lower endpoint positions) than the lower graph | |
| Name: accuracy | Type: double |
| Description: current accuracy of visualization | |
| Name: prevAccuracies | Type: ArrayList<Double> |
| Description: previous accuracies of classes no longer being visualized | |
| Name: allDataChecked | Type: boolean |
| Description: whether to display the all data confusion matrix | |
| Name: withoutOverlapChecked | Type: boolean |
| Description: whether to display the data without overlap confusion matrix | |
| Name: overlapChecked | Type: boolean |
| Description: whether to display the overlap confusion matrix | |
| Name: worstCaseChecked | Type: boolean |
| Description: whether to display the worst case confusion matrix | |
| Name: hasID | Type: boolean |
| Description: whether the input data has an ID column for the first column | |
| Name: hasClasses | Type: boolean |
| Description: whether the input data has a class column for the last column | |
| Name: zScoreMinMax | Type: boolean |
| Description: whether to perform a z-sore min-max normalization or a min-max normalized | |
| Name: angles | Type: double[] |
| Description: angle for every feature in a datapoint | |
| Name: prevAngles | Type: double[] |
| Description: previous angles before optimizing | |
| Name: data | Type: ArrayList<DataObject> |
| Description: holds normalized input data for each class | |
| Name: originalData | Type: ArrayList<DataObject> |
| Description: holds original input data for each class | |
| Name: allClasses | Type: ArrayList<String> |
| Description: class for every datapoint in dataset | |
| Name: uniqueClasses | Type: ArrayList<String> |
| Description: unique classes in dataset | |
| Name: classNumber | Type: int |
| Description: total number of classes | |
| Name: fieldNames | Type: ArrayList<String> |
| Description: list of each feature name in a dataset | |
| Name: fieldLength | Type: int |
| Description: total number of features (dimensions) in a dataset | |

**Functions/Methods:**

|  |  |
| --- | --- |
| Name: DV | Params: void |
| Return type: N/A | |
| Description: Constructor for DV Program. Creates main panel, menu bar, toolbar, sliders, and graphs. | |
| Name: createMenuBar | Params: void |
| Return type: void | |
| Description: Creates menu bar. Uses: creating and saving projects, importing data, and info popups. | |
| Name: createToolBar | Params: void |
| Return type: void | |
| Description: Creates tool bar. Uses: graph color control, visualization options, confusion matrix toggle, screen reset, visualization optimization, undo optimization, and bar graph toggle. | |
| Name: uiPanel | Params: void |
| Return type: void | |
| Description: Creates main panel. Holds graphs and sliders. | |
| Name: blankGraph | Params: void |
| Return type: void | |
| Description: creates a blank graph | |
| Name: createNewProject | Params: void |
| Return type: void | |
| Description: gets user input then creates new visualization project | |
| Name: importData | Params: void |
| Return type: void | |
| Description: imports additional data to current project | |
| Name: openSavedProject | Params: void |
| Return type: void | |
| Description: loads previously saved project | |
| Name: saveProject | Params: void |
| Return type: void | |
| Description: saves current project | |
| Name: normalizationInfoPopup | Params: void |
| Return type: void | |
| Description: displays popup giving information on different normalization techniques used | |
| Name: resetProgram | Params: void |
| Return type: void | |
| Description: removes current project | |

**RangeSlider**

**RangeSliderUI**

**ThresholdSliderUI**

**Resolutions**

**VisOptionsMenu**

**Python Files:**