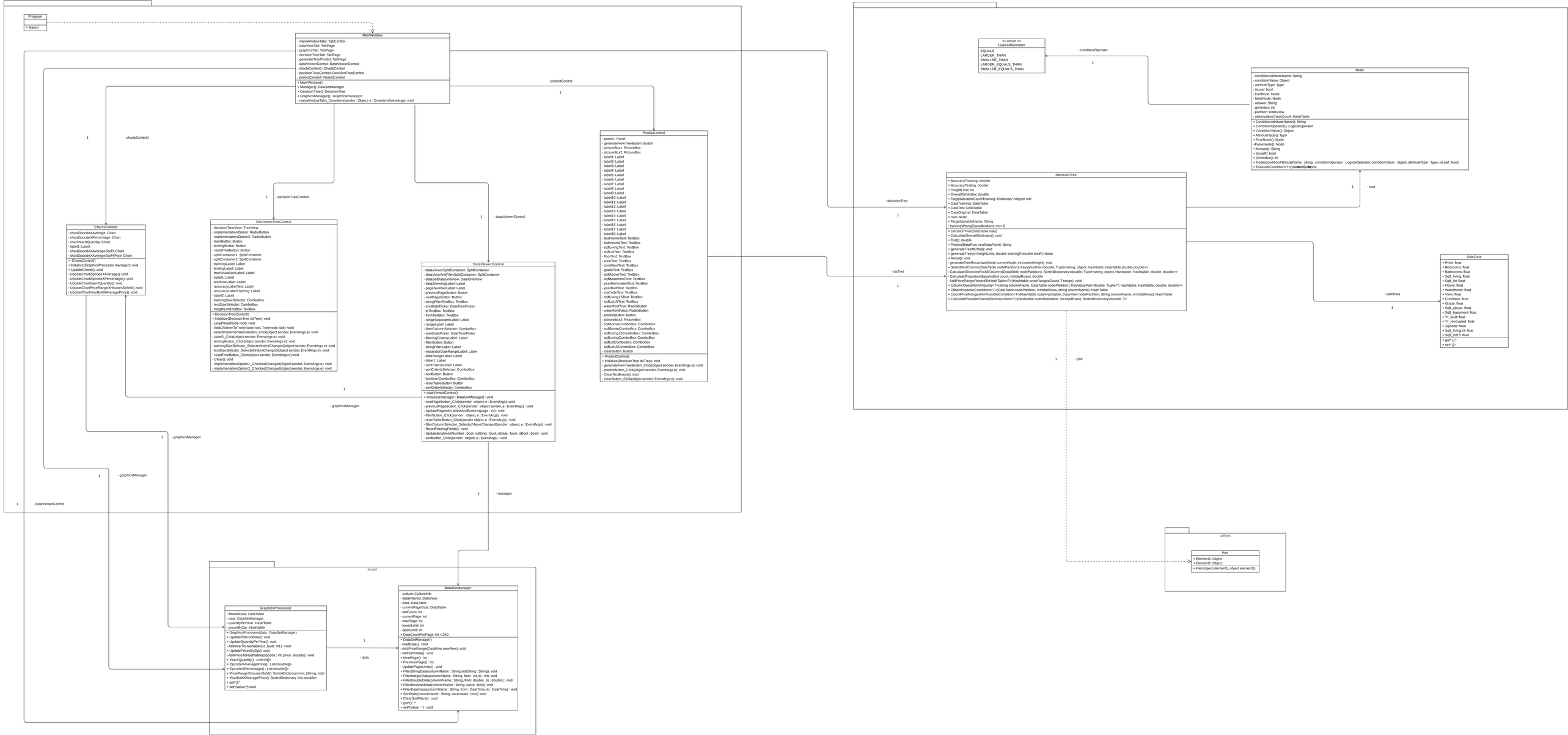
Program
+ Main()
MainWindow
- mainWindowTabs: TabControl
- dataViewTab: TabPage
- graphicsTab: TabPage
- decisionTreeTab: TabPage
- generateTreePredict: TabPage
- dataViewerControl: DataViewerControl
- chartsControl1: ChartsControl
- decisionTreeControl: DecisionTreeControl
- predictControl: PredictControl
+ MainWindow()
+ Manager(): DataSetManager
+ DecisionTree(): DecisionTree
+ GraphicsManager(): GraphicsProcessor
- mainWindowTabs_DrawItem(sender : Object, e : DrawItemEventArgs): void
- predictControl
1
1
- chartsControl1
1
- decisionTreeControl
1
- dataViewerControl
PredictControl
- panel1: Panel
- generateNewTreeButton: Button
- pictureBox1: PictureBox
- pictureBox2: PictureBox
- label1: Label
- label2: Label
- label3: Label
- label4: Label
- label5: Label
- label6: Label
- label7: Label
- label8: Label
- label9: Label
- label10: Label
- label11: Label
- label12: Label
- label13: Label
- label14: Label
- label15: Label
- label16: Label
- label17: Label
- label18: Label
- bedroomsText: TextBox
- bathroomsText: TextBox
- sqftLivingText: TextBox
- sqftLotText: TextBox
- floorText: TextBox
- viewText: TextBox
- conditionText: TextBox
- gradeText: TextBox
- sqftAboveText: TextBox
- sqftBasementText: TextBox
- yearRenovatedText: TextBox
- yearBuiltText: TextBox
- zipCodeText: TextBox
- sqftLiving15Text: TextBox
- sqftLot15Text: TextBox
- waterfrontTrue: RadioButton
- waterfrontFalse: RadioButton
- predictButton: Button
- pictureBox3: PictureBox
- sqftAboveComboBox: ComboBox
- sqftBsmtComboBox: ComboBox
- sqftLiving15ComboBox: ComboBox
- sqftLivingComboBox: ComboBox
- sqftLotComboBox: ComboBox
- sqftLot15ComboBox: ComboBox
- clearButton: Button
+ PredictControl()
+ Initialize(DecisionTree dsTree): void
- generateNewTreeButton_Click(object sender, EventArgs e): void
- predictButton_Click(object sender, EventArgs e): void
- ClearTextBoxes(): void
- clearButton_Click(object sender, EventArgs e): void
ChartsControl
- chart2pcodeXAverage: Chart
- chart2pcodeXPercentage: Chart
- chart1YearXQuantity: Chart
- label1: Label
- chart2pcodeXAverageSqFt: Chart
- chart2pcodeXAverageSqFtPrice: Chart
+ ChartsControl()
+ Initialize(GraphicsProcessor manager): void
+ UpdateCharts(): void
- UpdateChart2pcodeXAverage(): void
- UpdateChart2pcodeXPercentage(): void
- UpdateChart1YearXQuantity(): void
- UpdateChartPriceRangeXHousesSelled(): void
- UpdateChartYearBuiltXAveragePrice(): void
DecisionTreeControl
- decisionTreeView: TreeView
- implementationOption: RadioButton
- implementationOption2: RadioButton
- trainButton: Button
- testingButton: Button
- resetTreeButton: Button
- splitContainer1: SplitContainer
- splitContainer2: SplitContainer
- trainingLabel: Label
- testingLabel: Label
- treeVisualizerLabel: Label
- label1: Label
- testSizeLabel: Label
- accuracyLabelTest: Label
- accuracyLabelTraining: Label
- label2: Label
- trainingSizeSelector: ComboBox
- testSizeSelector: ComboBox
- heightLimitTxtBox: TextBox
+ DecisionTreeControl()
+ Initialize(DecisionTree dsTree): void
- LoadTreeNode root): void
- AddChildrenToTreeNode root, TreeNode dad): void
- selectImplementationButton_Click(object sender, EventArgs e): void
- label2_Click(object sender, EventArgs e): void
- testingButton_Click(object sender, EventArgs e): void
- trainingSizeSelector_SelectedIndexChanged(object sender, EventArgs e): void
- testSizeSelector_SelectedIndexChanged(object sender, EventArgs e): void
- resetTreeButton_Click(object sender, EventArgs e):void
- Clear(): void
- implementationOption1_CheckedChanged(object sender, EventArgs e): void
- implementationOption2_CheckedChanged(object sender, EventArgs e): void
DataViewerControl
- dataViewerSplitContainer: SplitContainer
- dataViewAndFilterSplitContainer: SplitContainer
- dataSetDataGridView: DataGridView
- dataShowingLabel: Label
- pageNumberLabel: Label
- prevousPageButton: Button
- nextPageButton: Button
- stringFilterTextBox: TextBox
- endDatePicker: DateTimePicker
- toTextBox: TextBox
- fromTextBox: TextBox
- rangeSeparatorLabel: Label
- rangeLabel: Label
- filterColumnSelector: ComboBox
- startDatePicker: DateTimePicker
- filteringCriteriaLabel: Label
- filterButton: Button
- stringFilterLabel: Label
- separatorDateRangeLabel: Label
- dateRangeLabel: Label
- label1: Label
- sortCriteriaLabel: Label
- sortCriteriaSelector: ComboBox
- sortButton: Button
- booleanComboBox: ComboBox
- resetTableButton: Button
- sortOrderSelector: ComboBox
+ dataViewerControl()
+ Initialize(manager : DataSetManager) : void
- nextPageButton_Click(sender : object, e : EventArgs): void
- prevousPageButton_Click(sender : object sender, e : EventArgs) : void
- UpdatePageInfoLabelsAndButtons(page : int) : void
- filterButton_Click(sender : object, e : EventArgs) : void
- resetTableButton_Click(sender object, e : EventArgs) : void
- filterColumnSelector_SelectedValueChanged(sender : object, e : EventArgs) : void
- ResetFilteringFields() : void
- UpdateEnableb(isNumber : bool, isString : bool, isDate : bool, isBool : bool) : void
- sortButton_Click(sender : object, e : EventArgs) : void
- graphicsManager
1
- graphicsManager
1
- graphicsManager
1
- dataViewerControl
1
- manager
Model
GraphicsProcessor
- filteredData: DataTable
- data: DataSetManager
- quantityPerYear: HashTable
- pricesByZip : Hashtable
+ GraphicsProcessor(data : DataSetManager)
+ UpdateFilteredData() : void
+ UpdateQuantityPerYear() : void
- AddYearToHashtable(yr_built : int) : void
+ UpdatePricesByZip() : void
- AddPriceToHashtable(zipcode : int, price : double) : void
+ YearXQuantity() : List<int[]>
+ ZpcodeXAveragePrice() : List<double[]>
+ ZpcodeXPercentage() : List<double[]>
+ PriceRangeXHousesSoldS(): SortedDictionary<int,(String, int)>
+ YearBuiltXAveragePrice(): SortedDictionary <int, double>
+ get*():*
+ set*(value:*):void
1
- data
DataSetManager
- culture: CultureInfo
- dataFiltered: DataView
- data: DataTable
- currentPageData: DataTable
- datCount: int
- currentPage: int
- maxPage: int
- lowerLimit: int
- operLimit: int
+ DataCountPerPage: int = 250
+ DataSetManager()
- loadData() : void
- AddPriceRange(DataRow newRow): void
- RefreshData() : void
+ NextPage() : int
+ PrevousPage() : int
- UpdatePageLimits() : void
+ FilterStringData(columnName : String,subString : String): void
+ FilterIntegerData(columnName : String, from : int, to : int): void
+ FilterDoubleData(columnName : String, from: double , to : double) : void
+ FilterBooleanData(columnName : String, value : bool): void
+ FilterDateData(columnName : String, from : DateTime, to : DateTime) : void
+ SortData(columnName : String, ascendant : bool): void
+ ClearSortFilters() : void
+ get*() : *
+ set*(value : *) : void
- predictControl
1
- dsTree
1
- decisionTree
1
- pair
1
- conditionOperator
1
- root
1
- saleData
1
<< enum >>
LogicalOperator
EQUALS
LARGER_THAN
SMALLER_THAN
LARGER_EQUALS_THAN
SMALLER_EQUALS_THAN
Node
- conditionAttributeName: String
- conditionValue: Object
- attributeType: Type
- isLeaf: bool
- trueNode: Node
- falseNode: Node
- answer: String
- giniIndex: int
- partition: DataView
- observationClassCount: HashTable
+ ConditionAttributeName(): String
+ ConditionOperator(): LogicalOperator
+ ConditionValue(): Object
+ AttributeType(): Type
+ TrueNode(): Node
+FalseNode(): Node
+ Answer(): String
+ IsLeaf(): bool
+ GiniIndex(): int
+ Node(conditionAttributeName : string , conditionOperator : LogicalOperator, conditionValue : object, attributeType : Type, isLeaf: bool)
+ EvaluateCondition<T>(value : T, node : Node)
DecisionTree
+ AccuracyTraining: double
+ AccuracyTesting: double
+ HeightLimit: int
+ OverallGiniIndex: double
+ TargetVariableCountTraining: Dictionary <object, int>
+ DataTraining: DataTable
+ DataTest: DataTable
+ DataOriginal: DataTable
+ root: Node
+ TargetVariableName: String
- trainingWrongClassifications: int = 0
+ DecisionTree(DataTable data)
+ CalculateOverallGiniIndex() : void
+ Test() : double
+ Predict(DataRow newDataPoint): String
+ generateTreeMLNet() : void
+ generateTree(int heightLimit, double trainingP, double testP): Node
+ Reset() : void
- generateTreeRecursive(Node currentNode, int currentHeight): void
+ SelectBestColumn(DataTable nodePartition): KeyValuePair<double, Tuple<string, object, Hashtable, Hashtable,double,double>>
- CalculateGiniIndexForAllColumns(DataTable nodePartition): SortedDictionary<double, Tuple<string, object, Hashtable, Hashtable, double, double>>
- CalculateProportionSquared(int count, int totalRows): double
- AddPriceRangeRecordToHashTable<T>(Hashtable priceRangersCount, T range): void
+ ColumnOverallGiniImpurity<T>(string columnName, DataTable nodePartition): KeyValuePair<double, Tuple<T, Hashtable, Hashtable, double, double>>
+ ObtainPossibleConditions<T>(DataTable nodePartition, int totalRows, string columnName): HashTable
+ CountPriceRangesForPossibleCondition<T>(Hashtable outerHashtable, DataView nodePartition, String columnName, int totalRows): HashTable
+ CalculatePossibleOverallGiniImpurities<T>(Hashtable outerHashtable, int totalRows): SortedDictionary<double, T>
SaleData
+ Price: float
+ Bedrooms: float
+ Bathrooms: float
+ Sqft_living: float
+ Sqft_lot: float
+ Floors: float
+ Waterfronts: float
+ View: float
+ Condition: float
+ Grade: float
+ Sqft_above: float
+ Sqft_basement: float
+ Yr_built: float
+ Yr_renovated: float
+ Zipcode: float
+ Sqft_living15: float
+ Sqft_lot15: float
+ get* () :*
+ set* (): *
Utilities
Pair
+ Element1: Object
+ Element2: Object
+ Pair(object element1, object element2)