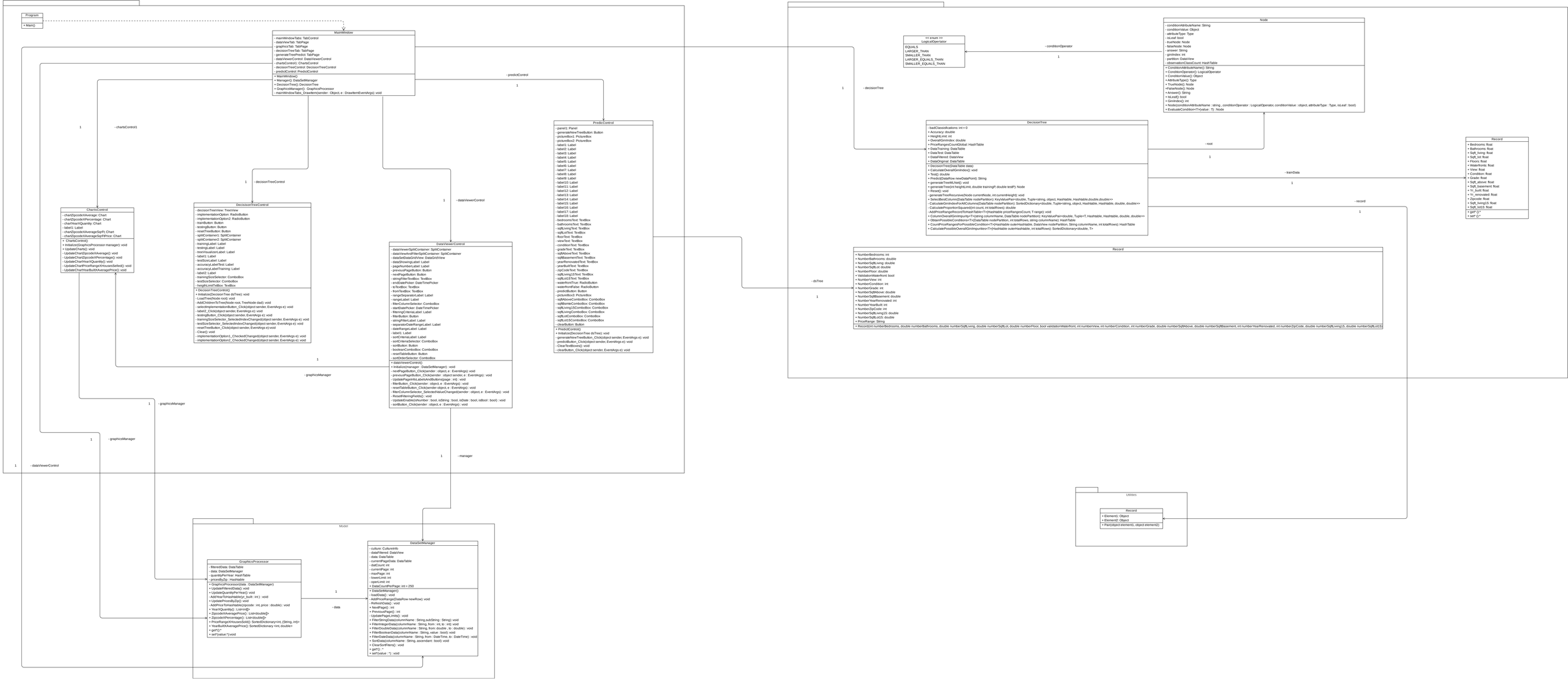
Program
+ Main()
MainWindow
- mainWindowTabs: TabControl
- dataViewTab: TabPage
- graphicsTab: TabPage
- decisionTreeTab: TabPage
- generateTreePredict: TabPage
- dataViewerControl: DataViewerControl
- chartsControl1: ChartsControl
- decisionTreeControl: DecisionTreeControl
- predictControl: PredicControl
+ MainWindow()
+ Manager(): DataSetManager
+ DecisionTree(): DecisionTree
+ GraphicsManager(): GraphicsProcessor
- mainWindowTabs_DrawItem(sender: Object, e : DrawItemEventArgs): void
- predictControl
1
PredicControl
- 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
- waterfrontFalse: RadioButton
- waterfrontTrue: RadioButton
- predictButton: Buton
- pictureBox3: PictureBox
- sqftAboveComboBox: ComboBox
- sqftBasementComboBox: ComboBox
- sqftLiving15ComboBox: ComboBox
- sqftLivingComboBox: ComboBox
- sqftLotComboBox: ComboBox
- sqftLot15ComboBox: ComboBox
- clearButton: Button
+ PredicControl()
+ 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
- chartsControl1
1
ChartsControl
- chartZipcodeAverage: Chart
- chartZipcodePercentage: Chart
- chartYearXQuantity: Chart
- label1: Label
- chartZipcodeAverageSqFt: Chart
- chartZipcodeAverageSqFtPrice: Chart
+ ChartsControl()
+ Initialize(GraphicsProcessor manager): void
+ UpdateCharts(): void
- UpdateChartZipcodeXAverage(): void
- UpdateChartZipcodeXPercentage(): void
- UpdateChartYearXQuantity(): void
- UpdateChartPriceRangeXHousesSelled(): void
- UpdateChartYearBuiltXAveragePrice(): void
- decisionTreeControl
1
DecisionTreeControl
- decisionTreeTreeView: TreeView
- implementationOption: RadioButton
- implementationOption2: RadioButton
- trainButton: Button
- testingBtutton: Buton
- resetTreeButton: Button
- splitContainer1: SplitContainer
- splitContainer2: SplitContainer
- treeLabel: Label
- trainingLabel: Label
- treeVisualizerLabel: Label
- label1: Label
- testSizeLabel: Label
- accuracyLabelTest: Label
- accuracyLabelTraining: Label
- label2: Label
- trainingSizeSelector: ComboBox
- testSizeSelector: ComboBox
- heightLimitTeBox: TextBox
+ DecisionTreeControl()
+ Initialize(DecisionTree dsTree): void
- LoadTree(Node root): void
- AddChildrenToTree(Node root, TreeNode dad): void
- selectImplementationButton_Click(object sender, EventArgs e): void
- label2_Click(object sender, EventArgs e): void
- testingBtutton_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
1
DataViewerControl
- dataViewerSplitContainer: SplitContainer
- dataViewAndFilterSplitContainer: SplitContainer
- dataSetDataGridView: DataGridView
- dataShowingLabel: Label
- pageNumberLabel: Label
- previousPageButton: Button
- nextPageButton: Button
- stringFilterTextBox: TextBox
- endDatePicker: DateTimePicker
- toTreeBox: TextBox
- fromTextBox: TextBox
- rangeSeparatorLabel: Label
- rangeLabel: Label
- filterColumnSelector: ComboBox
- startDatePicker: DateTimePicker
- filterColumnLabel: Label
- filterButton: Button
- stringFilterLabel: Label
- separatorDateRangeLabel: Label
- dateRangeLabel: Label
- labelt1: 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
- previousPageButton_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
- UpdateEnableUNumber : bool, isString : bool, isDate : bool, isBool : bool) : void
- sortButton_Click(sender : object, e : EventArgs) : void
- dsTree
1
- dataViewerControl
1
- graphicsManager
1
- graphicsManager
1
- graphicsManager
1
- manager
1
Model
GraphicsProcessor
- filteredData: DataTable
- data: DataSetManager
- quantityPerYear: HashTable
- priceByZip : Hashtable
+ GraphicsProcessor(data : DataSetManager)
+ UpdateFilteredData() : void
+ UpdateQuantityPerYear() : void
- AddYearToHashtable(yr_built : int) : void
+ UpdatePricesByZip() : void
- AddPriceToHashtable(zipcode : int, price : double) : void
+ YearXQuantity() : List<int[]>
+ ZipcodeXAveragePrice() : List<double[]>
+ ZipcodeXPercentage() : List<double[]>
+ PriceRangeXHousesSold() : SortedDictionary<int,(String,,int)>
+ YearBuiltXAveragePrice() : SortedDictionary <int, double>
+ get*();*
+ set*(value:*):void
- data
1
DataSetManager
- culture: CultureInfo
- dataFiltered: DataView
- data: DataTable
- currentPageData: DataTable
- datCount: int
- currentPage: int
- maxPage: int
- lowerLimit: int
- upperLimit: int
+ DataCountPerPage: int = 250
+ DataSetManager()
- loadData() : void
- AddPriceRangeOfDataRow newRow): void
- RefreshData() : void
+ NextPage() : int
+ PreviousPage() : 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
DecisionTree
- badClassifications: int = 0
+ Accuracy: double
+ HeightLimit: int
+ OverallGimindex: double
+ PriceRangesCountGlobal : HashTable
+ DataTraining: DataTable
+ DataTest: DataTable
+ DataFiltered: DataView
+ DataOriginal: DataTable
+ DecisionTree(DataTable data)
+ CalculateOverallGimindex() :void
+ Test(): double
+ Predict(DataRow newDataPoint): String
+ generateTreeML.Net() :void
+ generateTree(int heightLimit, double trainingPt, double testPt): Node
+ Reset() :void
- generateTreeRecursive(Node currentNode, int currentHeight): void
+ SelectBestColumn(DataTable nodePartition): KeyValuePair<double, Tuple<string, object, Hashtable, Hashtable,double,double>>
- CalculateGimindexForAllColumns(DataTable nodePartition): SortedDictionary<double, Tuple<string, object, Hashtable, Hashtable, double, double>>
- CalculateProportionSquared(int count, int totalRows): double
- AddPriceRangeRecordToHashTable<T>(Hashtable priceRangesCount, T range): void
+ ColumnOverallGiminpurity<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
+ CalculatePossibleOverallGiminpurities<T>(Hashtable outerHashtable, int totalRows): SortedDictionary<double, T>
- decisionTree
1
- root
1
- trainData
1
- record
1
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)
+ EvaluateConditionT<value : T> : Node
- conditionOperator
1
<< enum >>
LogicalOperator
EQUALS
LARGER_THAN
SMALLER_THAN
LARGER_EQUALS_THAN
SMALLER_EQUALS_THAN
Record
+ 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*()*
Record
+ NumberBedrooms: int
+ NumberBathrooms: double
+ NumberSqftLiving: double
+ NumberSqftLot: double
+ NumberFloor: double
+ ValidationWaterfront: bool
+ NumberView: int
+ NumberCondition: int
+ NumberGrade: int
+ NumberSqftAbove: double
+ NumberSqftBasement: double
+ NumberYearRenovated: int
+ NumberYearBuilt: int
+ NumberZipCode: int
+ NumberSqftLiving15: double
+ NumberSqftLot15: double
+ PriceRange: String
+ Record(int numberBedrooms, double numberBathrooms, double numberSqftLiving, double numberSqftLot, double numberFloor, bool validationWaterfront, int numberView, int numberCondition, int numberGrade, double numberSqftAbove, double numberSqftBasement, int numberYearRenovated, int numberZipCode, double numberSqftLiving15, double numberSqftLot15)
Utilities
Record
+ Element1: Object
+ Element2: Object
+ Pair(object element1, object element2)