EuclideanDistance

+ compute(a: Instance, b: Instance, weights: List<Float>): float

<<Interface>>
DistanceMetric

+ compute(a: Instance, b: Instance, weights: List<Float>): float

ManhattanDistance

+ compute(a: Instance, b: Instance, weights: List<Float>): float

ChebyshevDistance

+ compute(a: Instance, b: Instance, weights: List<Float>): float

Da...

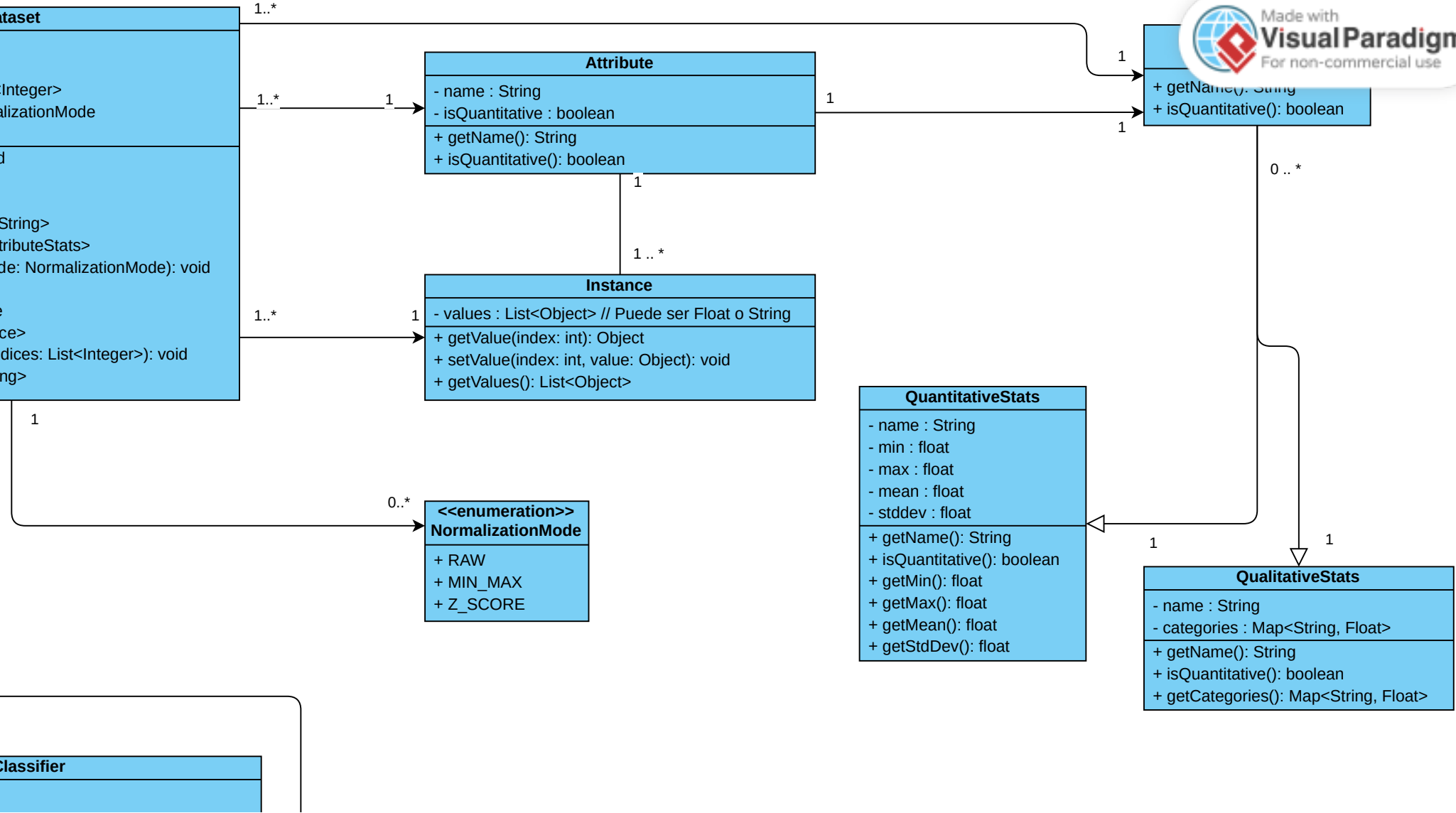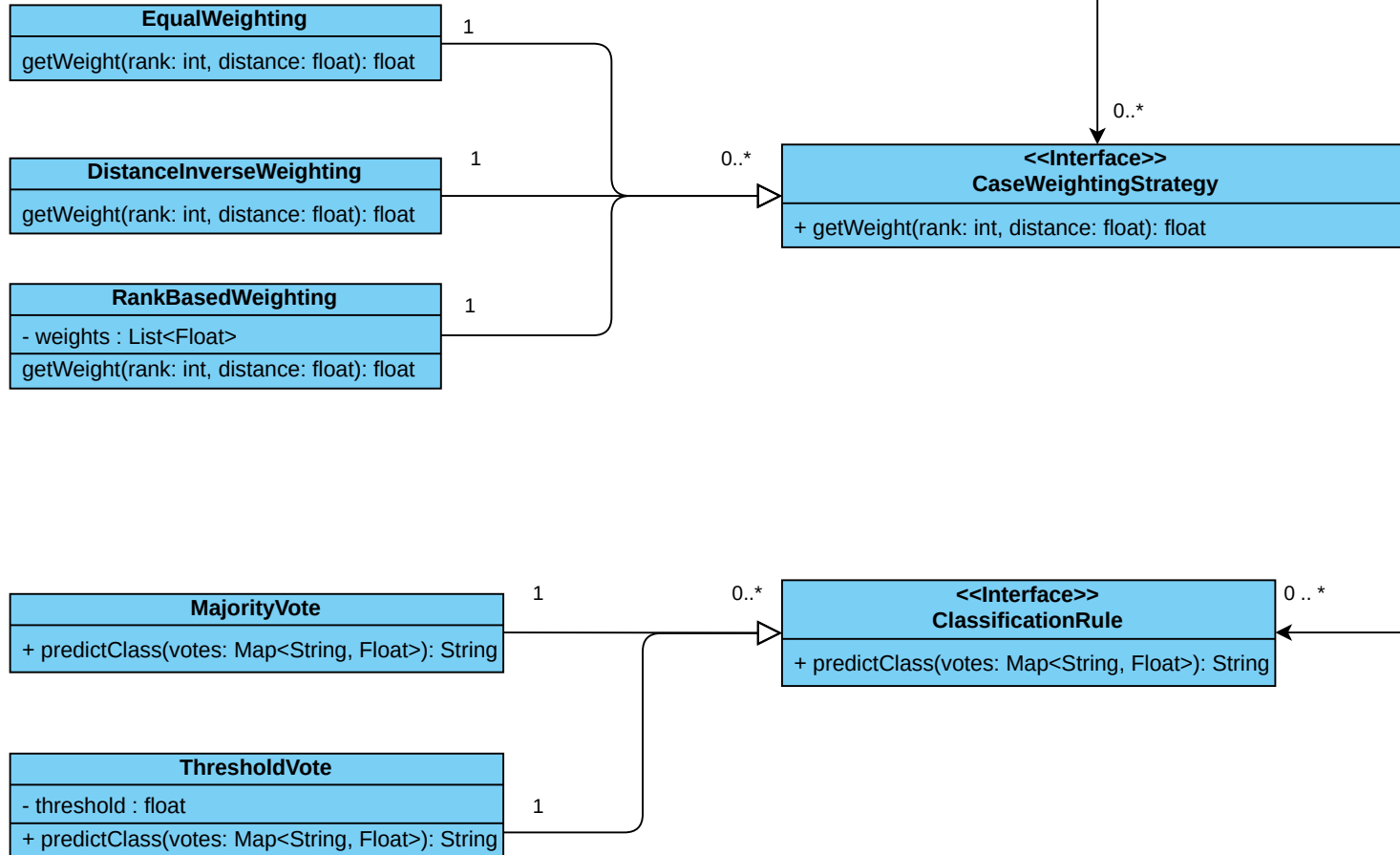- classAttributeIndices : List<...
- normalizationMode : Norma...
- weights : List<Float>

+ loadCSV(path: String): voi...
+ getNumAttributes(): int
+ getNumInstances(): int
+ getAttributeNames(): List<S...
+ getAttributeStats(): List<Att...
+ setNormalizationMode(mo...
+ normalize(): void
+ getInstance(i: int): Instanc...
+ getInstances(): List<Instan...
+ setClassAttributeIndices(in...
+ getClassValues(): List<Stri...

KNN...

+ k: int

1   0..*   1   0..*   0 .. *   1   0 .. *   0..*

Attribute

- name : String
- isQuantitative : boolean

+ getName(): String
+ isQuantitative(): boolean

Instance

- values : List<Object> // Puede ser Float o String

+ getValue(index: int): Object
+ setValue(index: int, value: Object): void
+ getValues(): List<Object>

QuantitativeStats

- name : String
- min : float
- max : float
- mean : float
- stddev : float

+ getName(): String
+ isQuantitative(): boolean
+ getMin(): float
+ getMax(): float
+ getMean(): float
+ getStdDev(): float

QualitativeStats

- name : String
- categories : Map<String, Float>

+ getName(): String
+ isQuantitative(): boolean
+ getCategories(): Map<String, Float>

<<enumeration>>
NormalizationMode

+ RAW
+ MIN_MAX
+ Z_SCORE

+ getName(): String
+ isQuantitative(): boolean

Classifier

1..*

1..*   1

1..*   1

1

1

1

1 .. *

1

0.. *

0..*

1

1

1

EqualWeighting

getWeight(rank: int, distance: float): float

DistanceInverseWeighting

getWeight(rank: int, distance: float): float

RankBasedWeighting

- weights : List<Float>

getWeight(rank: int, distance: float): float

**<<Interface>>
CaseWeightingStrategy**

+ getWeight(rank: int, distance: float): float

MajorityVote

+ predictClass(votes: Map<String, Float>): String

ThresholdVote

- threshold : float

+ predictClass(votes: Map<String, Float>): String

**<<Interface>>
ClassificationRule**

+ predictClass(votes: Map<String, Float>): String

- distanceMetric : DistanceMetri
...ightir

...e): S
+ setK(k: int): void
+ setDistanceMetric(dm: Distanc
+ setCaseWeighting(cw: CaseW
+ setClassificationRule(rule: Cla

Exp

- dataset : Dataset
- testSet : List<Instance>
- trainSet : List<Instance>
+ splitData(testRatio: float, rand
+ saveSplit(pathTrain: String, pa
+ loadSplit(pathTrain: String, pa
+ evaluate(knn: KNNClassifier):
+ getConfusionMatrix(knn: KNN

1

1

1

0..*

0..*

1

0..*

0 .. *

1

0..*

0 .. *

c
gStrategy

tring

ceMetric): void
WeightingStrategy): void
assificationRule): void

0..*

1   1

perimentManager

om: boolean, seed: int): void
athTest: String): void
thTest: String): void
float
Classifier): Map<String, Map<String, Integer>>