Esercitazione 5 – RTTI, Serializzazione

RTTI - Modellazione di un attributo continuo

Per modellare un attributo continuo (ad esempio la temperatura può variare in un range da 0° a 30° in maniera continua), implemetare la classe ContinuousNode e modificare di conseguenza la classe data, che contiene l'apprendimento del training set, e la classe RegressionTree, che contiene la modellazione di un albero di regressione, in modo tale che essa possa gestire, in maniera dinamica con l'uso dell'RTTI, sia attributi discreti che attributi continui.

Classe ContinuousNode

Implementare la classe ContinuousNode. La classe rappresenta un nodo corrispondente ad un attributo continuo. Implementa la classe SplitNode, quindi implentare tutti i metodi astratti di questa:

public ContinuousNode(Data trainingSet,int beginExampelIndex, int endExampleIndex, ContinuousAttribute attribute)

void setSplitInfo(Data trainingSet,int beginExampleIndex, int endExampleIndex, Attribute attribute)

fornita dal docente

int testCondition (Object value)

public String toString()

Classe RegressionTree

Ridefinire il metodo determineBestSplitNode() usando l'**RTTI** per distinguere un nodo discreto da un nodo continuo.

```
private SplitNode determineBestSplitNode(Data trainingSet,int begin,int end)

{

        ...

        for(int i=0;i<trainingSet.getNumberOfExplanatoryAttributes();i++){

        Attribute a=trainingSet.getExplanatoryAttribute(i);

        if(a instanceof DiscreteAttribute){

        DiscreteAttribute attribute=(DiscreteAttribute)trainingSet.getExplanatoryAttribute(i);

        currentNode = new DiscreteNode(trainingSet,begin,end,attribute);

        }

        else
```

```
        {
        ...
        }
        ts.add(currentNode);
        }
        …
}
```

Se necessario modificare i metodi **public Double predictClass(), public void printRules()** e **private void printRules(String current)** in modo da sfruttare l'**RTTI** per riconoscere un nodo foglia.

Classe Data

Da modificare il costruttore per trattare dati numerici. I metodi per la realizzazione dell'ordinamento anche in base ad attributi numerici sono forniti dal docente (quicksort.txt)

Esempio di dataset con attributi numerici: provaC.dat

Esempio di Output

Training set:

provaC.dat

Starting data acquisition phase!

Starting learning phase!

********* RULES **********

X=A AND Y<=2.0 ==> Class=1.0

X=A AND Y>2.0 ==> Class=1.5

X=B ==> Class=10.0

****

********* TREE **********

DISCRETE SPLIT : attribute=X Nodo: [Examples:0-14] variance:255.83333333333331 Split Variance: 0.625

child 0 split value=A[Examples:0-9]

child 1 split value=B[Examples:10-14]

CONTINUOUS SPLIT : attribute=Y Nodo: [Examples:0-9] variance:0.625 Split Variance: 0.0

child 0 split value<=2.0[Examples:0-4]

child 1 split value>2.0[Examples:5-9]

LEAF : class=1.0 Nodo: [Examples:0-4] variance:0.0

LEAF : class=1.5 Nodo: [Examples:5-9] variance:0.0

LEAF : class=10.0 Nodo: [Examples:10-14] variance:0.0

****

Starting prediction phase!

0:X=A

1:X=B

0

0:Y<=2.0

1:Y>2.0

0

1.0

Would you repeat ? (y/n)

y

Starting prediction phase!

0:X=A

1:X=B

1

10.0

Would you repeat ? (y/n)

y

Starting prediction phase!

0:X=A

1:X=B

0

0:Y<=2.0

1:Y>2.0

1

1.5

Would you repeat ? (y/n)

y

Starting prediction phase!

0:X=A

1:X=B

0

0:Y<=2.0

1:Y>2.0

3

tree.UnknownValueException: The answer should be an integer between 0 and 1!

Would you repeat ? (y/n)

n

Serializzazione

Si vuole dare la possibilità di salvare un albero appreso in un file di nome <NomeFileTraining>".dat" per utilizzarlo in seguito caricandolo da tale file.

Classe RegressionTree

Modificare la classe RegressionTree (e quelle ad essa aggregate direttamente e/o indirettamente) implementando l'interfaccia **Serializable**

Implementare i metodi **Salva** e **Carica**.

public void salva(String nomeFile) throws FileNotFoundException, IOException

Input: Nome del file in cui salvare l'albero

Output: Nessuno

Comportamento: Serializza l'albero in un file.

public static RegressionTree carica(String nomeFile) throws FileNotFoundException,IOException,ClassNotFoundException

Input: Nome del file in cui è salvato l'albero

Output: L'albero contenuto nel file.

Comportamento: Carica un albero di regressione salvato in un file.

Classe MainTest

Inserire nella classe **MainTest** **un menù di scelta per decidere se apprendere un albero di regressione da un training set e salvarlo su un file oppure caricare un albero di regressione precedentemente appreso dal file per predire nuovi esempi.**

Fornito dal Docente