03

Classificando uma instância

Transcrição

Agora já podemos começar a programar.

Para facilitar nossa vida, estará salvo na pasta "Banco 2" o arquivo "clientes.arff", base de dados que estamos tentando classificar e o arquivo "J48.model", modelo que usaremos para fazer a classificação. Nosso projeto estará dentro da pasta "eclipse-workspace" da máquina.

Vamos para o código agora. Precisaremos começá-lo com uma função `main()` e pressionaremos um "*Command + space*" depois para criá-la. Mas além disso, precisaremos de um `throws Exception` para lidar com algumas exceções criadas pelo Weka.

A ideia é criarmos um programa que pergunte para o usuário o nome do arquivo que ele quer classificar, pois pode haver mais de um arquivo. Como leremos a informação do usuário, antes de tudo vamos importar algumas bibliotecas. Primeiramente faremos o `import` da biblioteca `java.io.File` e posteriormente a `weka.core.Instances`, para ler algumas instâncias do arquivo. Leremos a informação fornecida pelo usuário, para isso, `java.util.Scanner` resolverá o problema.

Prosseguindo no corpo da função, perguntaremos para o usuário o nome do arquivo que devemos ler com um `System.out.println("Digite o nome do arquivo:")`. Pegaremos a informação digitada com `Scanner meuObj = new Scanner(System.in)`, sendo `(System.in)` a entrada do sistema.

Agora precisaremos guardar o nome do arquivo que o usuário digitar. faremos `String nomeArquivo = meuObj.nextLine()` e teremos que ler um arquivo `;arff`. Portanto faremos ainda outro `import`, o de `weka.core.converters.ArffLoader`.

Vamos efetivamente criar um `ArffLoader` para ver o arquivo, e diremos que ele `carrega = new ArffLoader()`. Carregaremos o DataSet com esse `carrega` e `setSource(newFile(nomeArquivo))`, ou seja, a fonte será o nome do arquivo, e passaremos esse nome para os parâmetros do método.

Tendo o arquivo `.arff`, precisaremos pegar instâncias dele com `Instances dado = carrega.getDataSet()`. Passaremos para o dado ainda qual atributo será a classe que queremos classificar. Pegaremos um número de atributos `-1`. Temos a última coluna do arquivo com a classe referente ao depósito,

Enfim, precisaremos importar também uma outra biblioteca para ler o modelo `weka.classifiers.trees.j48`. Feito isso, poderemos criar um `J48 arvore` e atribuir a ele `(J48) weka.core.SerializationHelper.read("j48.model")` e assim leremos efetivamente o arquivo que temos.

Precisaremos ainda classificar uma instância com `arvore.classifyInstance(dado.instance(1))`. "1" será o número da instância, dessa forma estaríamos classificando uma instância, por exemplo, a primeira instância do nosso arquivo. Isso servirá para classificar qualquer instância, contato que forneçamos o número da que queremos classificar dentro do arquivo.

```
import java.io.File;
import weka.core.Instances;
import java.util.Scanner;
```

```
import weka.core.converters.ArffLoader;
import weka.classifiers.trees.j48;

public class previsor {

    public static void main(String[] args) throws Exception{

        System.out.println("Digite o nome do arquivo:");

        Scanner meuObj = new Scanner(System.in);

        String nomeArquivo = meuObj.nextLine();

        ArffLoader carrega = new ArffLoader();

        carrega.getSource(new File(nomeArquivo));

        Instances dado = carrega.getDataSet();

        dado.setClassIndex(dado.numAttributes()-1);

        J48 arvore = (J48) weka. core.SerializationHelper.read("j48.model");

        arvore.classifyInstance(dado.instance(1));

        }

}
```