

Yapay Zeka ve Python Programlama

Emir Öztürk

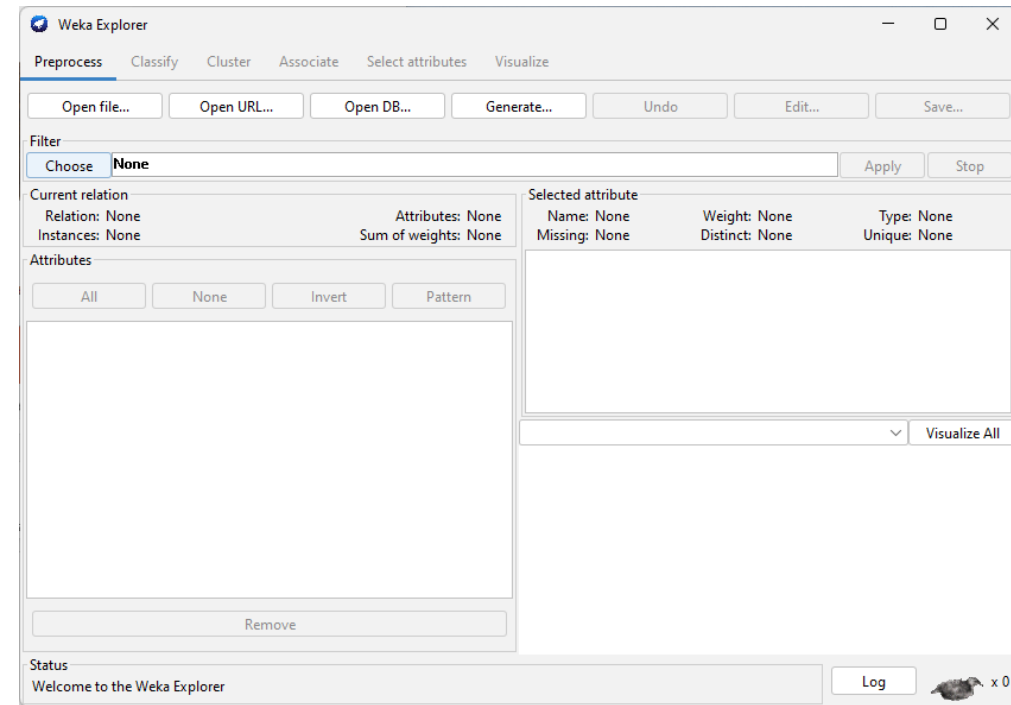
Weka

- Açık kaynak
- Veri işleme
- Makine öğrenmesi algoritmaları
- Görselleştirme



Weka

- Önişleme
- Sınıflandırma
- Kümeleme
- Atama
- Özellik seçimi



Arff format

- @Relation
- @Attribute
- @Data

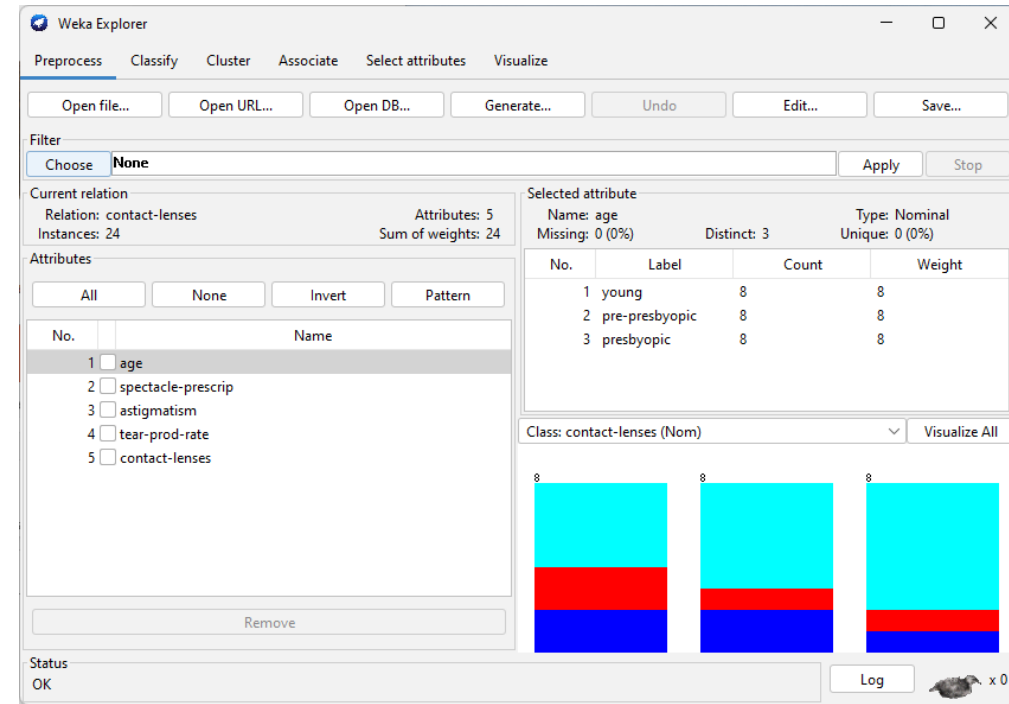
```
@relation 'sms-spam-dataset-w  
@attribute Text string  
@attribute Class {ham,spam}  
  
@data  
'Go until jurong point, crazy
```

```
@relation 'cpu'  
@attribute MYCT real  
@attribute MMIN real  
@attribute MMAX real  
@attribute CACH real  
@attribute CHMIN real  
@attribute CHMAX real  
@attribute class real  
@data  
125,256,6000,256,16,128,198  
29,8000,32000,32,8,32,269  
29,8000,32000,32,8,32,220
```

```
@relation contact-lenses  
  
@attribute age {young, pre-presbyopic, presbyopic}  
@attribute spectacle-prescrip {myope, hypermetrope}  
@attribute astigmatism {no, yes}  
@attribute tear-prod-rate {reduced, normal}  
@attribute contact-lenses {soft, hard, none}  
  
@data  
%  
% 24 instances  
%  
young,myope,no,reduced,none  
young,myope,no,normal,soft  
young,myope,yes,reduced,none  
young,myope,yes,normal,hard  
young,hypermetrope,no,reduced,none  
young,hypermetrope,no,normal,soft  
young,hypermetrope,yes,reduced,none  
young,hypermetrope,yes,normal,hard  
pre-presbyopic,myope,no,reduced,none  
pre-presbyopic,myope,no,normal,soft
```

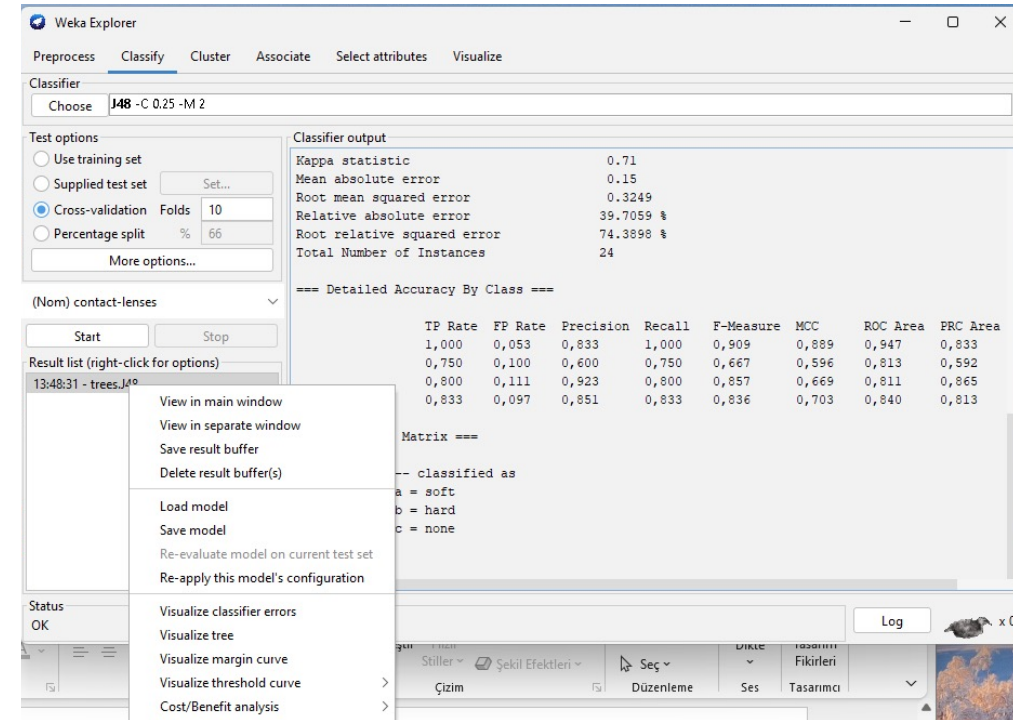
Önişleme

- Filtreleme
- Ayrıklaştırma
- Tip dönüşümü
- Tokenize etme



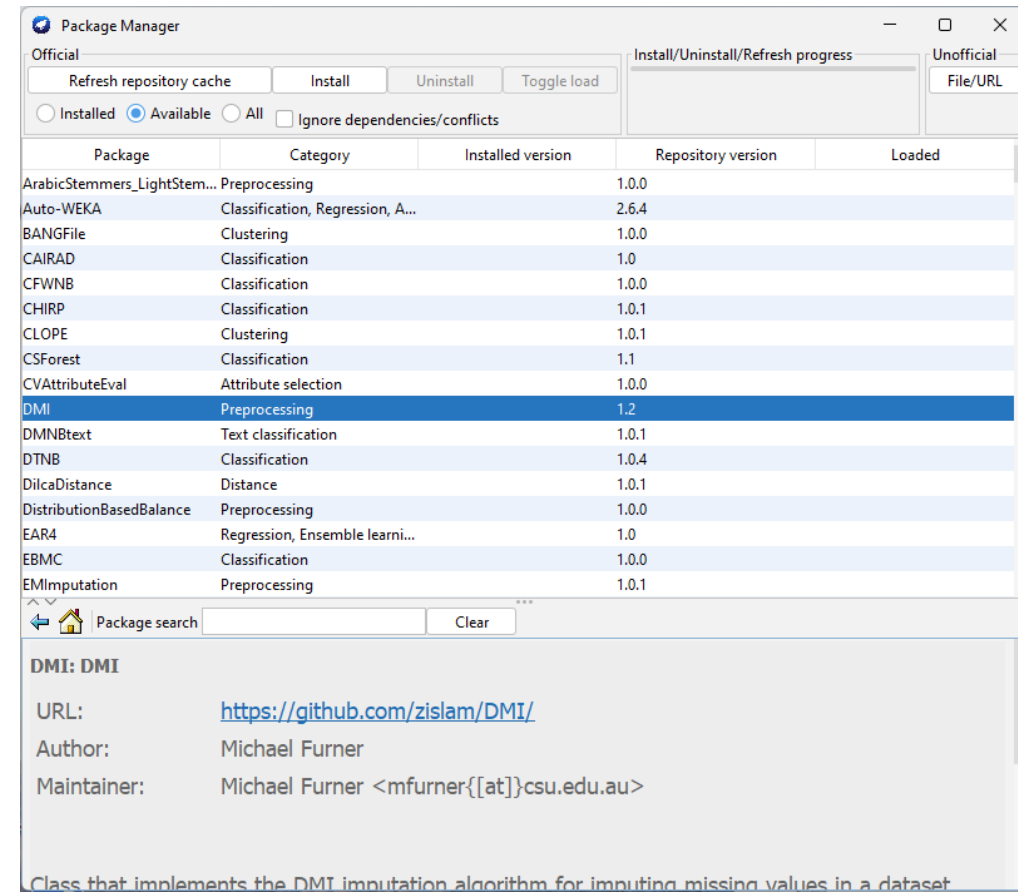
Sınıflandırma

- Eğitim seti
- Test seti
- Cross-validation
- Yüzdelik ile bölme
- Sınıflandırma algoritmaları
- Model oluşturma
- Sınıflandırma algoritmaları



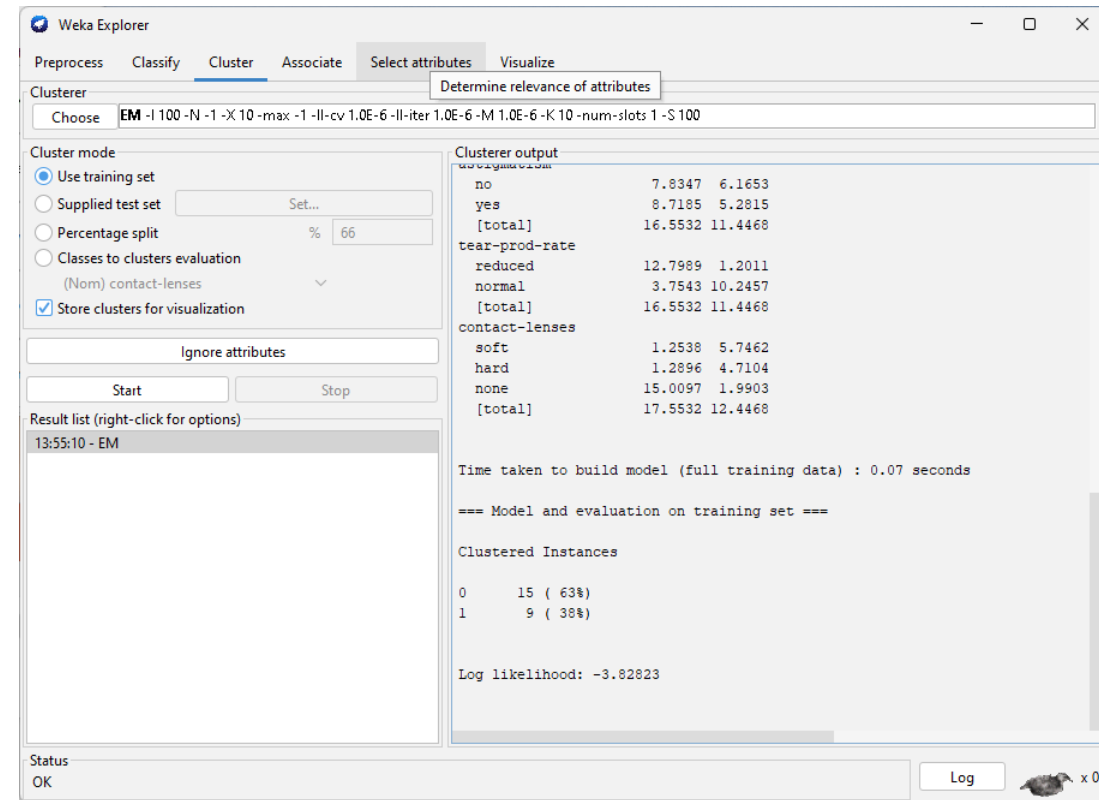
Algoritma – Paket eklenmesi

- Paket yöneticisi
- Kurulmak istenen paketlerin eklenmesi



Kümeleme

- Cross-validation yok
- Test seti ayrı verilebilir
- Model kaydı



Weka Explorer

Preprocess Classify **Cluster** Associate Select attributes Visualize

Determine relevance of attributes

Clusterer Choose **EM** -I 100 -N -1 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100

Cluster mode

- ☒ Use training set
- ☐ Supplied test set Set...
- ☐ Percentage split % 66
- ☐ Classes to clusters evaluation (Nom) contact-lenses
- ☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

13:55:10 - EM

Clusterer output

no	7.8347	6.1653
yes	8.7185	5.2815
[total]	16.5532	11.4468
tear-prod-rate		
reduced	12.7989	1.2011
normal	3.7543	10.2457
[total]	16.5532	11.4468
contact-lenses		
soft	1.2538	5.7462
hard	1.2896	4.7104
none	15.0097	1.9903
[total]	17.5532	12.4468

Time taken to build model (full training data) : 0.07 seconds

=== Model and evaluation on training set ===

Clustered Instances

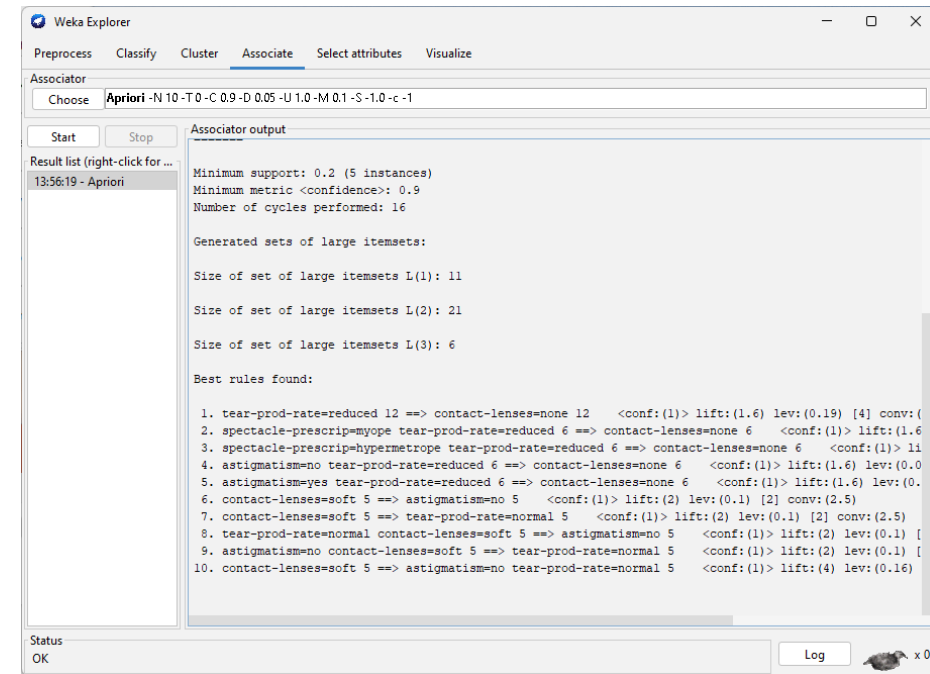
0	15	(63%)
1	9	(38%)

Log likelihood: -3.82823

Status OK Log x 0

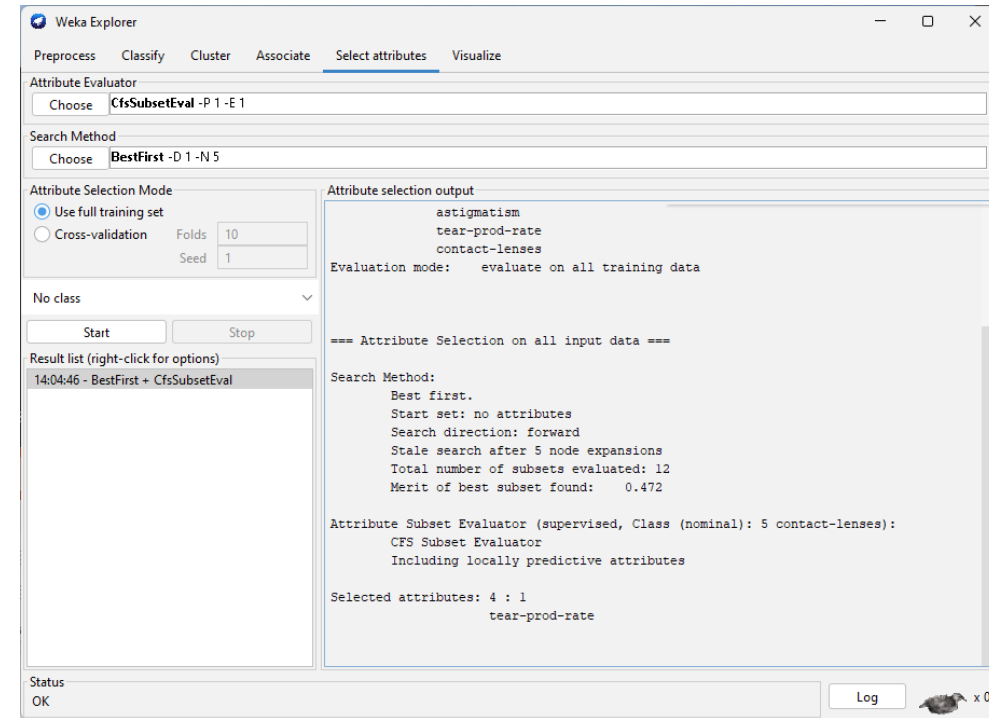
Atama

- Değişkenlerin ilişkilendirilmesi
- Belirli değişkenlerin birlikte yer aldığının tespiti



Özellik Seçimi

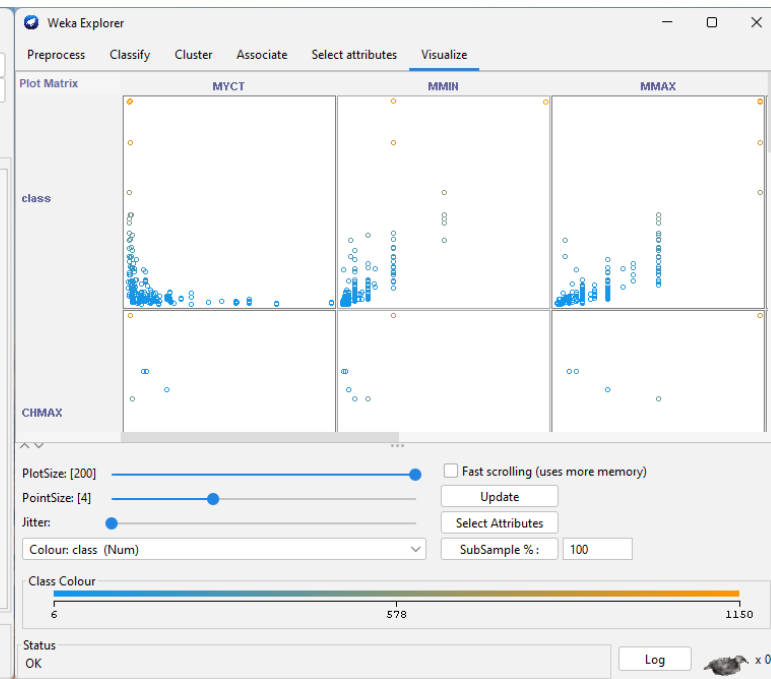
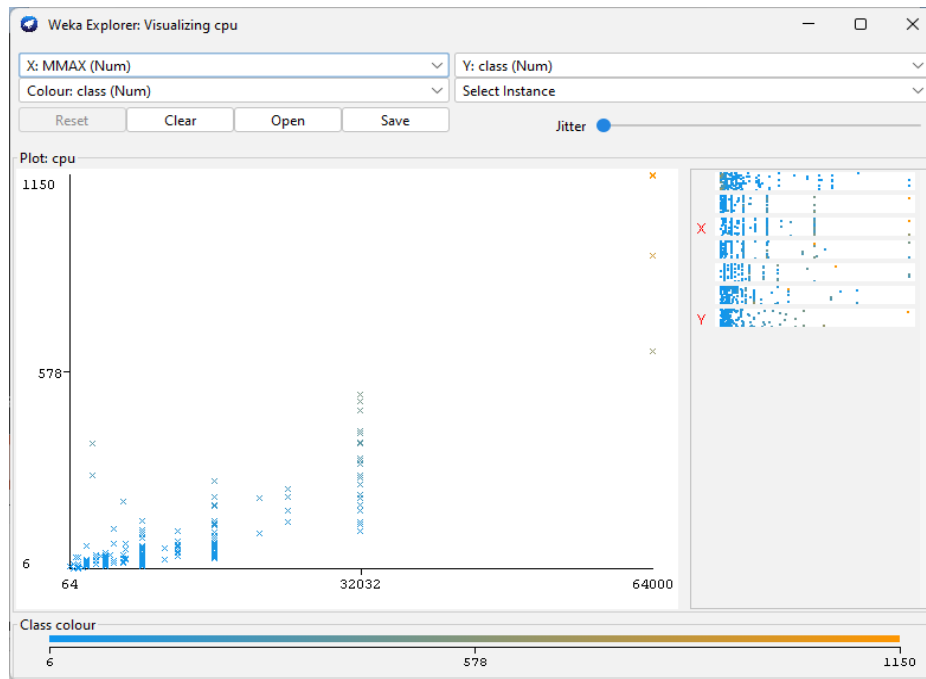
- Gereksiz özelliklerin elenmesi
- Her zaman iyileştirme gerçekleştirmez
- Farklı seçim algoritmaları



Görselleştirme

Veri dağılımının görselleştirilmesi

Farklı özelliklerin ikililer arasındaki ilişkisi



Model Testi

- Java
- Weka paketi
 - Maven
- Eğitim
- Test

```
public class Main {  
    public static void main(String[] args) throws Exception {  
        var instances : Instances = Train();  
        Test(instances);  
    }  
}  
  
1 usage  
private static void Test(Instances instances) throws Exception {  
    var model = (J48)SerializationHelper.read( filename: "C:\\Users\\emiro\\Desktop\\contact-lenses.model");  
    instances.add(new DenseInstance( numAttributes: 5));  
    instances.get(instances.size()-1).setValue( i: 0, s: "young");  
    instances.get(instances.size()-1).setValue( i: 1, s: "myope");  
    instances.get(instances.size()-1).setValue( i: 2, s: "no");  
    instances.get(instances.size()-1).setValue( i: 3, s: "reduced");  
    instances.get(instances.size()-1).setValue( i: 4, s: "soft");  
    double result = model.classifyInstance(instances.instance( index: instances.size()-1));  
    System.out.println(result);  
}  
  
1 usage  
private static Instances Train() throws Exception {  
    DataSource source = new DataSource( location: "C:\\Users\\emiro\\Desktop\\contact-lenses.arff");  
    Instances data = source.getDataSet();  
    data.setClassIndex(4);  
    J48 j48 = new J48();  
    j48.buildClassifier(data);  
    Evaluation evaluation = new Evaluation(data);  
    evaluation.crossValidateModel(j48,data, numFolds: 10,new Random( seed: 1));  
    System.out.println(evaluation.toSummaryString());  
    SerializationHelper.write( filename: "C:\\Users\\emiro\\Desktop\\contact-lenses.model",j48);  
    return data;  
}  
}
```