Yapay Zeka ve Python Programlama

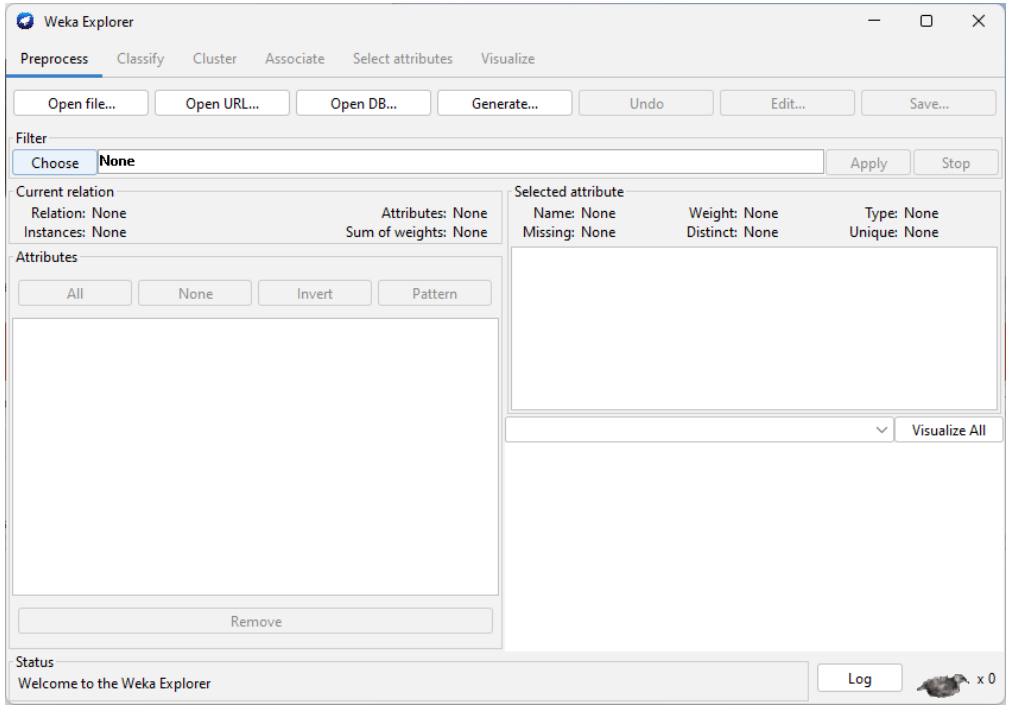
Emir Öztürk

Weka

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

Weka

- Önişleme
- Sınıflandırma
- Kümeleme
- Atama
- Özellik seçimi



Arff formatı

- @Relation
- @Attribute
- @Data

```
@relation 'sms-spam-dataset-w

@attribute Text string
@attribute Class {ham,spam}

@data
```

```
@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
```

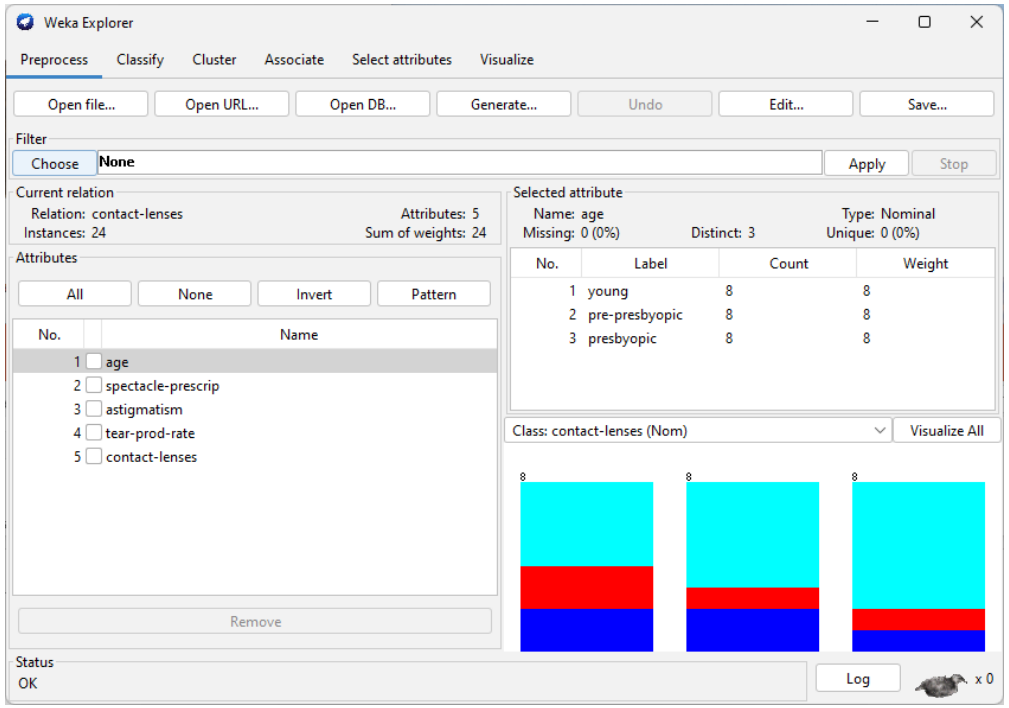
```
@relation contact-lenses

@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
```

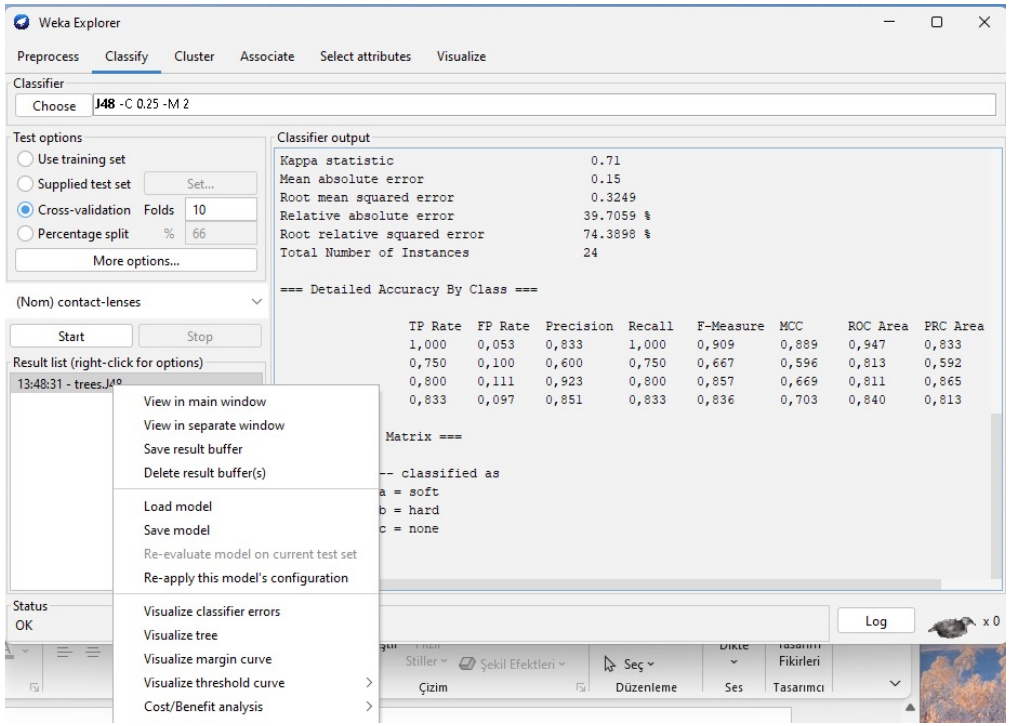
Önişleme

- Filtreleme
- Ayrıklaştırma
- Tip dönüşümü
- Tokenize etme



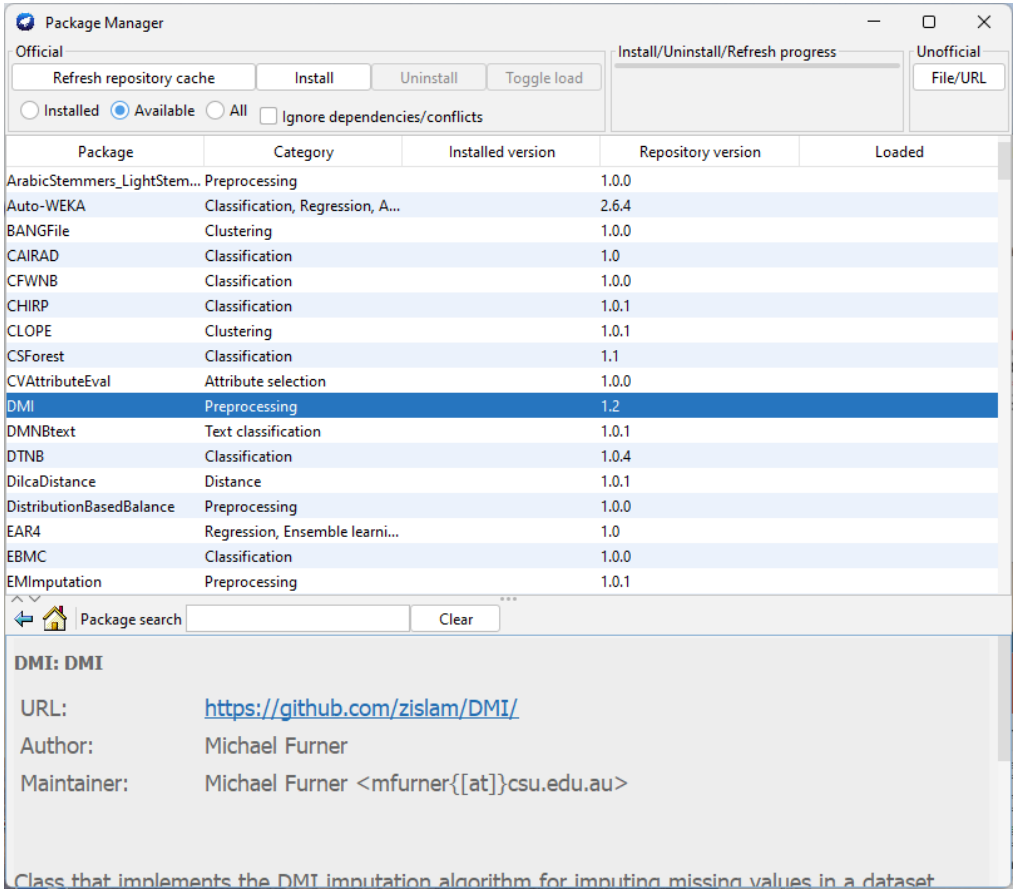
Sınıflandırma

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



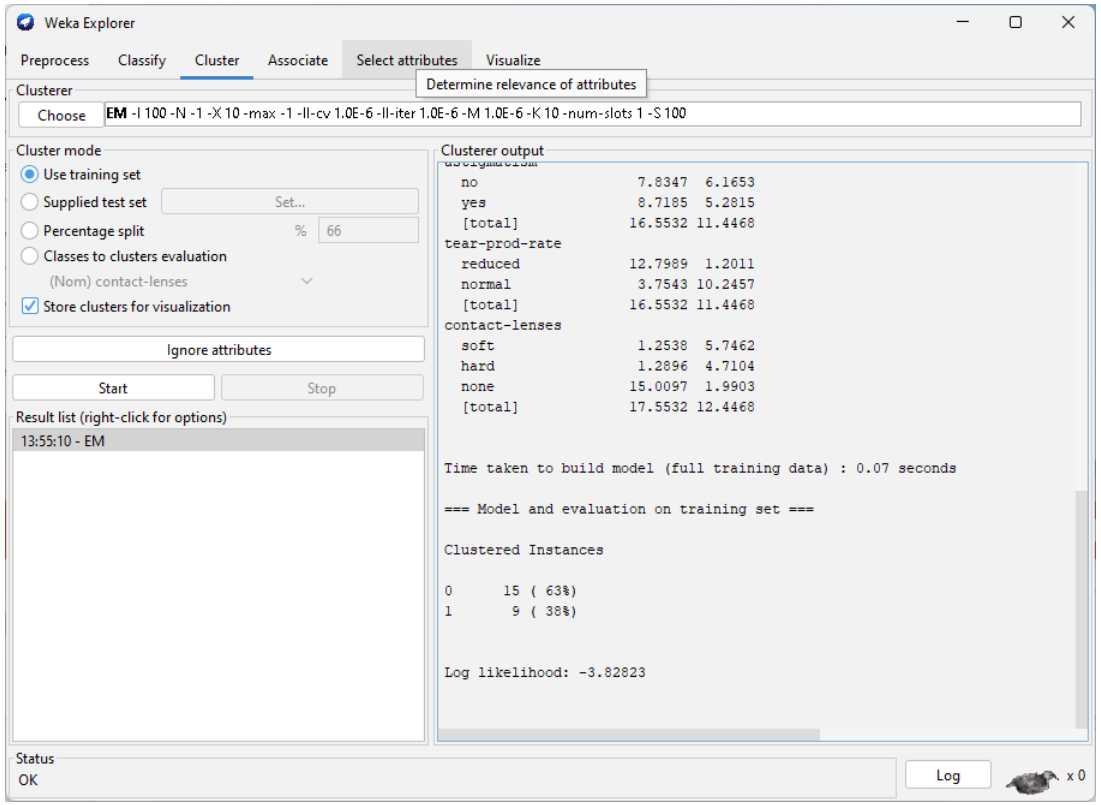
Algoritma – Paket eklenmesi

- Paket yöneticisi
- Kurulmak istenen paketlerin eklenmesi



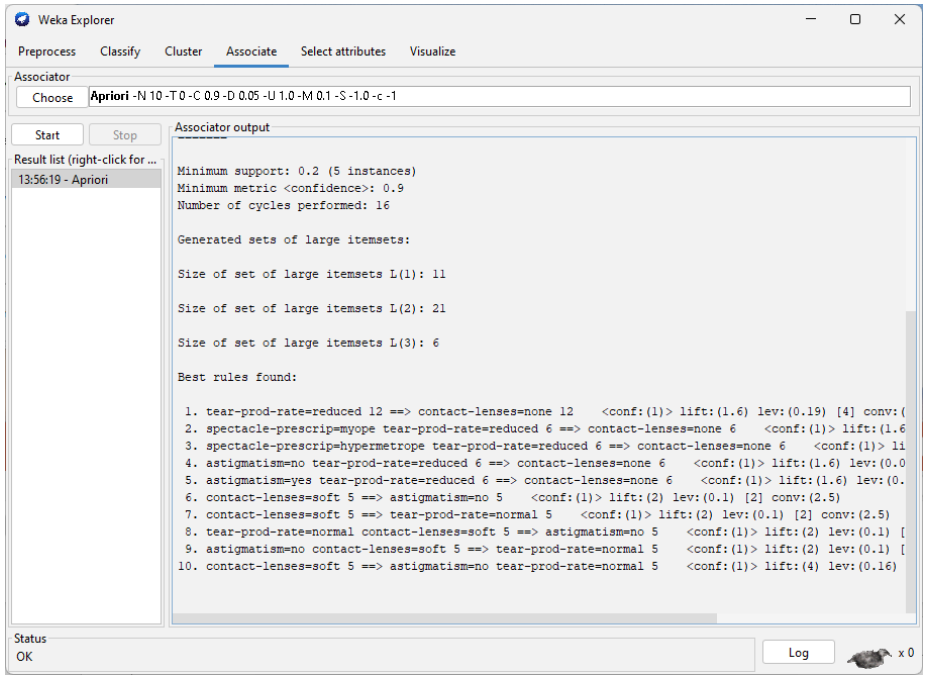
Kümeleme

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



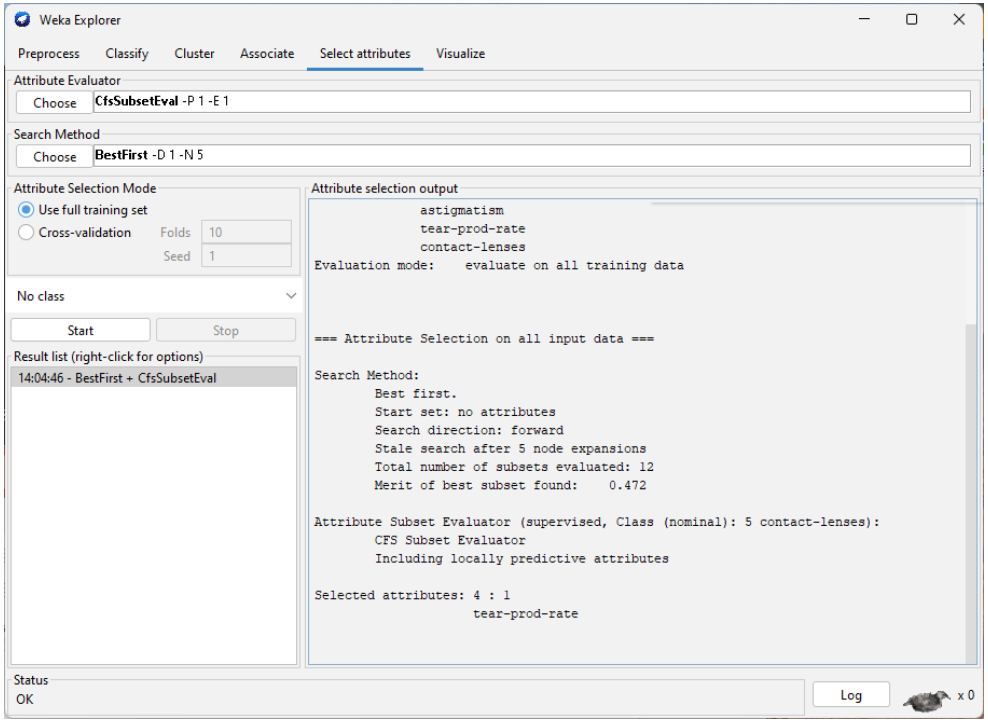
Atama

- Değişkenlerin ilişkilendirilmesi
- Belirli değişkenlerin birlikte yer aldığının tespiti



Özellik Seçimi

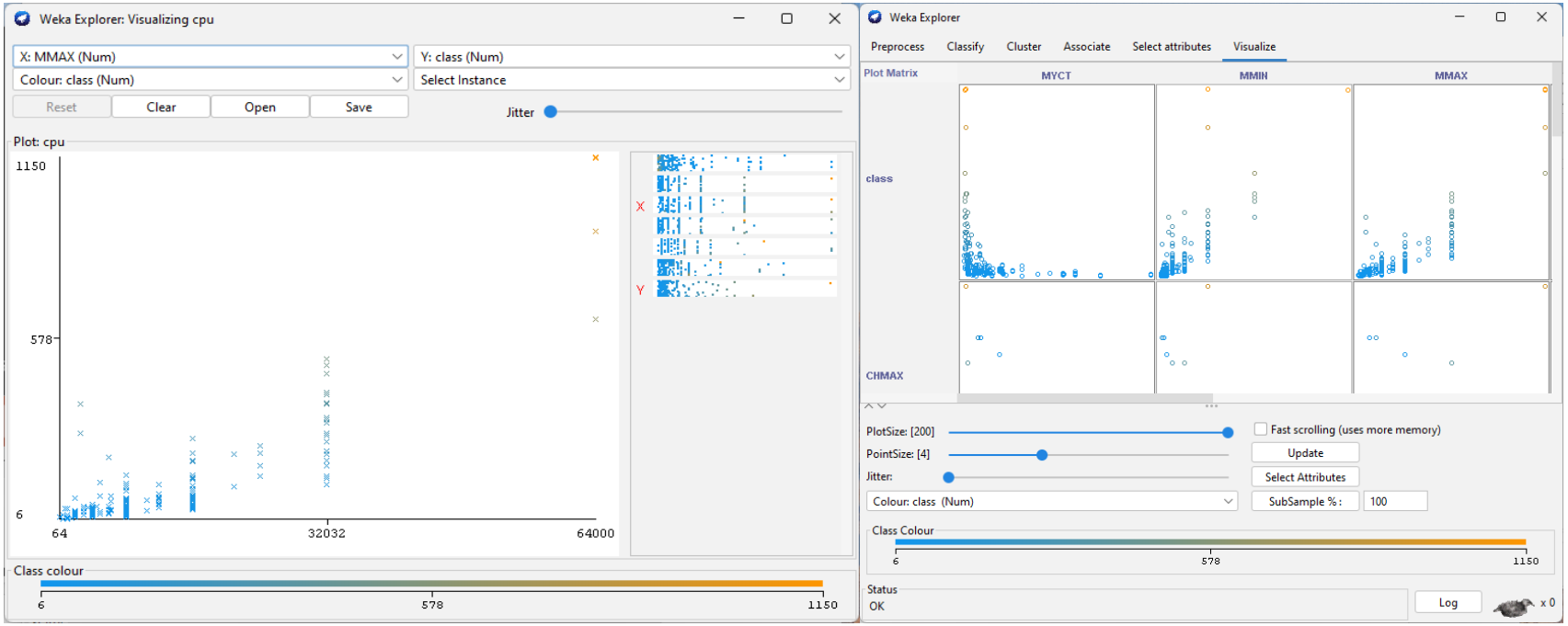
- Gereksiz özelliklerin elenmesi
- Her zaman iyileştirme gerçekleştirmez
- Farklı seçim algoritmaları



Görselleştirme

Veri dağılımının görselleştirilmesi

Farklı özelliklerin ikililer arasındaki ilişkisi



Model Testi

- Java
- Weka paketi
- Maven
- Eğitim
- Test

```java
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;
    }
}
```