

 免費電子書

學習

weka

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#weka

.....	1
<b>1: weka</b> .....	<b>2</b>
.....	2
Examples .....	2
.....	2
<b>Weka</b> .....	<b>2</b>
<b>2: WekaJython</b> .....	<b>3</b>
.....	3
.....	3
<b>wekaJython</b> .....	<b>3</b>
Examples .....	3
.....	3
.....	3
.....	4
.....	4
.....	5
.....	5
<b>3: Weka</b> .....	<b>7</b>
.....	7
.....	7
Examples .....	7
simpleCLIJython .....	7
<b>4: KnowledgeFlow</b> .....	<b>9</b>
.....	9
.....	9
TrainingSetMakerTestSetMaker .....	9
ArffSaver .....	9
KnowledgeFlowTimeSeriesForecasting .....	9
Examples .....	9
KnowledgeFlow .....	9
<b>5:</b> .....	<b>10</b>

Examples.....	10
ARFF.....	10
ARFF.....	10
<b>Weka &lt;3.5.5.....</b>	<b>10</b>
<b>Weka &gt;= 3.5.5.....</b>	<b>11</b>
.....	11
<b>6: WekaCPython.....</b>	<b>12</b>
.....	12
<b>WekaCPython.....</b>	<b>12</b>
Examples.....	12
Weka CPythonHello World.....	12
<b>7: WekaR.....</b>	<b>13</b>
.....	13
<b>WekaR.....</b>	<b>13</b>
<b>WekaR.....</b>	<b>13</b>
<b>Weka.....</b>	<b>13</b>
<b>R.....</b>	<b>14</b>
Examples.....	14
R.....	14
<b>8: .....</b>	<b>16</b>
Examples.....	16
LibLinear.....	16
.....	19

---

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [weka](#)

It is an unofficial and free weka ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official weka.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# 1: weka

weka。

weka。 weka。

## Examples

Weka。 Java。 Weka。 。

---

# Weka

WekaWeka 3.8Weka 3.9。 。

。 Weka 3.83.9WekaWeka。 。

Windows / MacOS / Linux。

WEKA

### pox.xml

```
<dependency>
  <groupId>nz.ac.waikato.cms.weka</groupId>
  <artifactId>weka-dev</artifactId>
  <version>3.9.1</version>
</dependency>
```

### gradle

```
compile group: 'nz.ac.waikato.cms.weka', name: 'weka-dev', version: '3.9.1'
```

[weka https://riptutorial.com/zh-TW/weka/topic/3699/weka](https://riptutorial.com/zh-TW/weka/topic/3699/weka)

---

## 2: WekaJython

**WekaJython** 1.ExplorerExperimenterKnowledgeFlowsimpleCLIWeka; 2.JythonWeka APIWeka; Python;

---

## wekaJython

1. Weka Package managerJythonJFreeChart;

2. nano .bash\_profile

3. .bash\_profile

```
export Weka_Data=User/Documents/Directory/Of/Your/Data
```

4.

5. source .bash\_profile

Weka tools Jython console

## Examples

```
# imports
import weka.core.converters.ConverterUtils.DataSource as DS
import weka.filters.Filter as Filter
import weka.filters.unsupervised.attribute.Remove as Remove
import os

# load data
data = DS.read(os.environ.get("MOOC_DATA") + os.sep + "iris.arff")

# remove class attribute
rem = Remove()
rem.setOptions(["-R", "last"])
rem.setInputFormat(data)
dataNew = Filter.useFilter(data, rem)

# output filtered dataset
print(dataNew)
```

```
# imports
import weka.core.converters.ConverterUtils.DataSource as DS
import weka.classifiers.trees.J48 as J48
import os

# load data
data = DS.read(os.environ.get("MOOC_DATA") + os.sep + "anneal.arff")
data.setClassIndex(data.numAttributes() - 1)

# configure classifier
cls = J48()
```

```
cls.setOptions(["-C", "0.3"])
```

```
# build classifier  
cls.buildClassifier(data)
```

```
# output model  
print(cls)
```

```
# imports  
import weka.core.converters.ConverterUtils.DataSource as DS  
import weka.classifiers.Evaluation as Evaluation  
import weka.classifiers.trees.J48 as J48  
import java.util.Random as Random  
import os  
  
# load data  
data = DS.read(os.environ.get("MOOC_DATA") + os.sep + "anneal.arff")  
data.setClassIndex(data.numAttributes() - 1)
```

```
# configure classifier  
cls = J48()  
cls.setOptions(["-C", "0.3"])
```

```
# cross-validate classifier  
evl = Evaluation(data)  
evl.crossValidateModel(cls, data, 10, Random(1))
```

```
# print statistics  
print(evl.toSummaryString("=== J48 on anneal (stats) ===", False))  
print(evl.toMatrixString("=== J48 on anneal (confusion matrix) ==="))
```

```
# imports  
import weka.classifiers.trees.J48 as J48  
import weka.core.converters.ConverterUtils.DataSource as DS  
import os
```

```
# load training data  
data = DS.read(os.environ.get("MOOC_DATA") + os.sep + "anneal_train.arff")  
data.setClassIndex(data.numAttributes() - 1)
```

```
# configure classifier  
cls = J48()  
cls.setOptions(["-C", "0.3"])
```

```
# build classifier on training data  
cls.buildClassifier(data)
```

```
# load unlabeled data  
dataUnl = DS.read(os.environ.get("MOOC_DATA") + os.sep + "anneal_unlbl.arff")  
dataUnl.setClassIndex(dataUnl.numAttributes() - 1)
```

```
# test compatibility of train/unlabeled datasets  
msg = dataUnl.equalHeadersMsg(data)  
if msg is not None:  
    print("train and prediction data are not compatible:\n" + msg)
```

```
# make predictions  
for inst in dataUnl:  
    dist = cls.distributionForInstance(inst)
```

```

labelIndex = cls.classifyInstance(inst)
label = dataUnl.classAttribute().value(int(labelIndex))
print(str(dist) + " - " + str(labelIndex) + " - " + label)

```

```

# Note: install jfreechartOffscreenRenderer package as well for JFreeChart library

```

```

# imports
import weka.classifiers.Evaluation as Evaluation
import weka.classifiers.functions.LinearRegression as LinearRegression
import weka.core.converters.ConverterUtils.DataSource as DS
import java.util.Random as Random
import org.jfree.data.xy.DefaultXYZDataset as DefaultXYZDataset
import org.jfree.chart.ChartFactory as ChartFactory
import org.jfree.chart.plot.PlotOrientation as PlotOrientation
import org.jfree.chart.ChartPanel as ChartPanel
import org.jfree.chart.renderer.xy.XYBubbleRenderer as XYBubbleRenderer
import org.jfree.chart.ChartUtilities as ChartUtilities
import javax.swing.JFrame as JFrame
import java.io.File as File
import os

```

```

# load data
data = DS.read(os.environ.get("MOOC_DATA") + os.sep + "bodyfat.arff")
data.setClassIndex(data.numAttributes() - 1)

```

```

# configure classifier
cls = LinearRegression()
cls.setOptions(["-C", "-S", "1"])

```

```

# cross-validate classifier
evl = Evaluation(data)
evl.crossValidateModel(cls, data, 10, Random(1))

```

```

# collect predictions
act = []
prd = []
err = []
for i in range(evl.predictions().size()):
    prediction = evl.predictions().get(i)
    act.append(prediction.actual())
    prd.append(prediction.predicted())
    err.append(abs(prediction.actual() - prediction.predicted()))

```

```

# create plot
plotdata = DefaultXYZDataset()
plotdata.addSeries("LR on " + data.relationName(), [act, prd, err])
plot = ChartFactory.createScatterPlot(
    "Classifier errors", "Actual", "Predicted", \
    plotdata, PlotOrientation.VERTICAL, True, True, True)
plot.getPlot().setRenderer(XYBubbleRenderer())

```

```

# display plot
frame = JFrame()
frame.setTitle("Weka")
frame.setSize(800, 800)
frame.setLocationRelativeTo(None)
frame.getContentPane().add(ChartPanel(plot))
frame.setVisible(True)

```



```

# imports
import weka.classifiers.bayes.BayesNet as BayesNet
import weka.core.converters.ConverterUtils.DataSource as DS
import weka.gui.graphvisualizer.GraphVisualizer as GraphVisualizer
import javax.swing.JFrame as JFrame
import os

# load data
data = DS.read(os.environ.get("MOOC_DATA") + os.sep + "iris.arff")
data.setClassIndex(data.numAttributes() - 1)

# configure classifier
cls = BayesNet()
cls.setOptions(["-Q", "weka.classifiers.bayes.net.search.local.K2", "--", "-P", "2"])

# build classifier
cls.buildClassifier(data)

# display tree
gv = GraphVisualizer()
gv.readBIF(cls.graph())
frame = JFrame("BayesNet - " + data.relationName())
frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE)
frame.setSize(800, 600)
frame.getContentPane().add(gv)
frame.setVisible(True)

# adjust tree layout
gv.layoutGraph()

```

WekaJython <https://riptutorial.com/zh-TW/weka/topic/8046/wekajython>

## 3: Weka

Weka Explorer KnowledgeFlow Experimenter Simple CLI Workbench

- 1.
- 2.

COS;

1. 53;
- 2.

### KnowledgeFlow

- 1.
- 2.

COS

1. KExperimenter [ADAMS](#);
2. KFWeka API;

### simpleCLI

pro Explorer

cos Weka API Jython Groovy

## Examples

### simpleCLI Jython

### simpleCLI

simpleCLI

```
java weka.classifiers.rules.ZeroR -t path/to/a-file-of-dataset
```

### Jython

#### Weka MOOC 5.1

```
# imports
import weka.core.converters.ConverterUtils.DataSource as DS
import weka.filters.Filter as Filter
import weka.filters.unsupervised.attribute.Remove as Remove
import os

# load data
data = DS.read(os.environ.get("MOOC_DATA") + os.sep + "iris.arff")
```

```
# remove class attribute
rem = Remove()
rem.setOptions(["-R", "last"])
rem.setInputFormat(data)
dataNew = Filter.useFilter(data, rem)

# output filtered dataset
print(dataNew)
```

**Weka** <https://riptutorial.com/zh-TW/weka/topic/8042/weka>

---

## 4: KnowledgeFlow

Weka KnowledgeFlowKF。 WekaKF。 。 WekaIstMark HallEibe FrankWeka

### TrainingSetMakerTestSetMaker

1. `ClassAssignerArffLoaderTrainingSetMakerTestSetMaker`。
- 

### ArffSaver

2. `arffArffSaverrelationNameForFilenameFalse`。
- 

### KnowledgeFlowTimeSeriesForecasting

1. `knowledgeFlowArffLoader`
  2. `ArffLoader`
  - 3.
  - 4.
  5. `ctrl + v`
  6. `ArffSaver`
- 

## Examples

### KnowledgeFlow

1. `.bash_profile`

```
function wekaflstart{
export R_HOME = / Library / Frameworks / R.framework / Resources
java -Xss10M -Xmx4096M -cpweka.jar weka.gui.knowledgeflow.KnowledgeFlow"$ 1"
}
```

2. `weka.jarwekastart "path to a knowledgeflow file"`

**KnowledgeFlow** <https://riptutorial.com/zh-TW/weka/topic/8053/knowledgeflow>

---

# 5:

## Examples

### ARFF

ARFF - Weka. ARFF.

- 
- 
- 
- 

```
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}
```

- 
- StringToWordVector◦
- 
- **JavaSimpleDateFormat ;ISO-8601**◦

```
@RELATION iris

@ATTRIBUTE sepallength NUMERIC
@ATTRIBUTE sepalwidth NUMERIC
@ATTRIBUTE petallength NUMERIC
@ATTRIBUTE petalwidth NUMERIC
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}
```

;?◦ ARFF

```
@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
```

### ARFF

WekaARFF.

---

## Weka <3.5.5

## ARFF

```
import weka.core.Instances;
import java.io.BufferedReader;
import java.io.FileReader;
...
BufferedReader reader = new BufferedReader(new FileReader("data.arff"));
Instances data = new Instances(reader);
reader.close();
data.setClassIndex(data.numAttributes() - 1);
```

◦ ARFF `data.numAttributes() - 1` ◦ Weka `buildClassifier` ◦

## Weka> = 3.5.5

WekaARFF ◦ CSVWeka ◦

```
import weka.core.converters.ConverterUtils.DataSource;
...
DataSource source = new DataSource("data.arff");
Instances data = source.getDataSet();
if (data.classIndex() == -1) {
    data.setClassIndex(data.numAttributes() - 1);
}
```

Weka ◦ DatabaseUtils.props; ◦

```
jdbcDriver=org.gjt.mm.mysql.Driver
jdbcURL=jdbc:mysql://localhost:3306/my_database
```

◦

```
import weka.core.Instances;
import weka.experiment.InstanceQuery;
...
InstanceQuery query = new InstanceQuery();
query.setUsername("user");
query.setPassword("pass");
query.setQuery("select * from mytable");
Instances data = query.retrieveInstances();
```

- JDBC ◦
- Microsoft AccessJDKJDBC-ODBC ◦
- InstanceQuery `VARCHARTEXT` ◦ `NominalToStringStringToNormal` ◦

<https://riptutorial.com/zh-TW/weka/topic/5928/>

---

# 6: WekaCPython

---

## WekaCPython

### wekaPython

1. tools package manager
2. wekaPython

### Python

1. anacondaconda
2. numpypandasmatplotlibscikit-learn
3. [conda](#)

## Examples

### Weka CPythonHello World

Explorer iris.arffCPython Scripting Python Scripts

```
hi = "Hello, CPython of Weka!"
hello = hi.upper()
iris = py_data
info = iris.describe()
```

Python Variables hi Get text

WekaCPython <https://riptutorial.com/zh-TW/weka/topic/7921/wekacpython->

---

## 7: WekaR.

---

### WekaR

1. R
  2. R
  3. WekaRWeka
- 

---

### WekaR.

#### Mac

1. Mark Hallinfo.Plist
2. R.
3. R<sub>rJava</sub>

`install.packages'rJava'`

4. Weka Package ManagerRplugin

5. weka 3-8-0

6. 2Michael Hall

`export R_HOME = / Library / Frameworks / R.framework / Resources  
java -Xss10M -Xmx4096M -cp weka.jar weka.gui.GUIChooser`

7. weka<sub>weka\_r.sh</sub>

8. `chmod a + x weka_r.sh`

9. `weka.jarweka 3-8-0`

`./weka_r.sh`

◦ `./weka_r.shWekaR.`

---

---

### Weka

#### Weka

Weka 3-8-0Weka 3-8-0



```
java -jar weka.jar
```

## Weka Explorer

1. preprocessopen file weka data folder;
2. R consoleR console boxR

## Weka KnowledgeFlow

1. Data mining processesDataSourcesArffLoader ;
2. ArffLoader
3. ScriptingRscriptExecutor
4. option +ArffLoader dataset RScript Executor
5. RScript ExecutorR
6. SettingsR ScriptingwekaR

---

## R

1. ExplorerKnowledgeFlowiris.arff ;
2. Plotting inside R ConsolePlotting inside R ConsolePlotting inside R Console

## Examples

### R

#### Weka

iris.arffwekaWeka ExplorerR consoleWeka KnowledgeFlowR Scripting

```
library(ggplot2)

ggplot(rdata, aes(x = petal.length)) + geom_density()

ggplot(rdata, aes(x = petal.length)) + geom_density() + xlim(0,8)

ggplot(rdata, aes(x = petal.length)) + geom_density(adjust = 0.5) + xlim(0,8)

ggplot(rdata, aes(x = petal.length, color = class)) + geom_density(adjust = 0.5) + xlim(0,8)

ggplot(rdata, aes(x = petal.length, color = class, fill = class)) + geom_density(adjust = 0.5)
+ xlim(0,8)

ggplot(rdata, aes(x = petal.length, color = class, fill = class)) + geom_density(adjust = 0.5,
alpha = 0.5) + xlim(0,8)

library(reshape2)
```

```
ndata = melt(rdata)
ndata

ggplot(ndata, aes(x = value, color = class, fill = class)) + geom_density(adjust = 0.5, alpha
= 0.5) + xlim(0,8) + facet_grid(variable ~ .)

ggplot(ndata, aes(x = value, color = class, fill = class)) + geom_density(adjust = 0.5, alpha
= 0.5) + xlim(0,8) + facet_grid(. ~ variable)

ggplot(ndata, aes(y = value, x = class, colour = class)) + geom_boxplot() + facet_grid(. ~
variable)
```

**WekaR.** <https://riptutorial.com/zh-TW/weka/topic/7916/wekar->

# 8:

## Examples

### LibLinear

- .arff

```
private static Instances getDataFromFile(String path) throws Exception{

    DataSource source = new DataSource(path);
    Instances data = source.getDataSet();

    if (data.classIndex() == -1){
        data.setClassIndex(data.numAttributes()-1);
        //last attribute as class index
    }

    return data;
}
```

```
Instances trainingData = getDataFromFile(pathToArffFile);
```

- **StringToWordVector**

\*

1. tf-idf
- 2.
- 3.
- 4.
5. n-gram\*

```
StringToWordVector() filter = new StringToWordVector();
filter.setWordsToKeep(1000000);
if(useIdf){
    filter.setIDFTransform(true);
}
filter.setTFTransform(true);
filter.setLowerCaseTokens(true);
filter.setOutputWordCounts(true);
filter.setMinTermFreq(minTermFreq);
filter.setNormalizeDocLength(new
SelectedTag(StringToWordVector.FILTER_NORMALIZE_ALL,StringToWordVector.TAGS_FILTER));
NGramTokenizer t = new NGramTokenizer();
t.setNGramMaxSize(maxGrams);
t.setNGramMinSize(minGrams);
filter.setTokenizer(t);
WordsFromFile stopwords = new WordsFromFile();
stopwords.setStopwords(new File("data/stopwords/stopwords.txt"));
filter.setStopwordsHandler(stopwords);
if (useStemmer){
```

```

    Stemmer s = new /*Iterated*/LovinsStemmer();
    filter.setStemmer(s);
}
filter.setInputFormat(trainingData);

```

- **trainingData** `trainingData = Filter.useFilter(trainingData, filter);`

- **LibLinear**

1. SVMType 0L2

2. `setProbabilityEstimates(true)` **probalities**

```

Classifier cls = null;
LibLINEAR liblinear = new LibLINEAR();
liblinear.setSVMTYPE(new SelectedTag(0, LibLINEAR.TAGS_SVMTYPE));
liblinear.setProbabilityEstimates(true);
// liblinear.setBias(1); // default value
cls = liblinear;
cls.buildClassifier(trainingData);

```

- 

```

System.out.println("Saving the model...");
ObjectOutputStream oos;
oos = new ObjectOutputStream(new FileOutputStream(path+"mymodel.model"));
oos.writeObject(cls);
oos.flush();
oos.close();

```

- `.arff`

```
Instances trainingData = getDataFromFile(pathToArffFile);
```

- 

```
Classifier myCls = (Classifier) weka.core.SerializationHelper.read(path+"mymodel.model");
```

- **StringToWordVectortestingData** `filter.setInputFormat(trainingData);` ◦

*InputMappedClassifier*

- **testingData** `testingData = Filter.useFilter(testingData, filter);`

- 1.

```

for (int j = 0; j < testingData.numInstances(); j++) {
    double res = myCls.classifyInstance(testingData.get(j));
}

```

`res` *double*. `arff` ◦ `testintData.classAttribute().value((int) res)`

---

2.

```
for (int j = 0; j < testingData.numInstances(); j++) {  
    double[] dist = first.distributionForInstance(testInstances.get(j));  
}
```

*dist.arff*

- *myClassifier.setProbabilityEstimates(true); myClassifier.setProbabilityEstimates(true);*

<https://riptutorial.com/zh-TW/weka/topic/7753/>

---

S. No		Contributors
1	weka	<a href="#">Community</a> , <a href="#">Gal Dreiman</a>
2	WekaJython	<a href="#">Daniel</a>
3	Weka	<a href="#">Daniel</a>
4	KnowledgeFlow	<a href="#">Daniel</a>
5		<a href="#">SJB</a>
6	WekaCPython	<a href="#">Daniel</a>
7	WekaR.	<a href="#">Daniel</a>
8		<a href="#">xro7</a>