

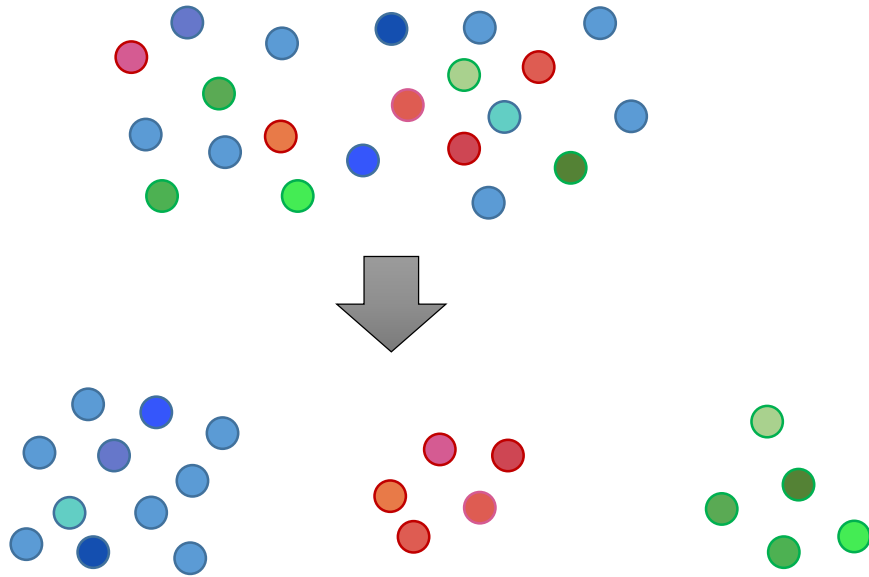
Introduction to Machine Learning



Topics

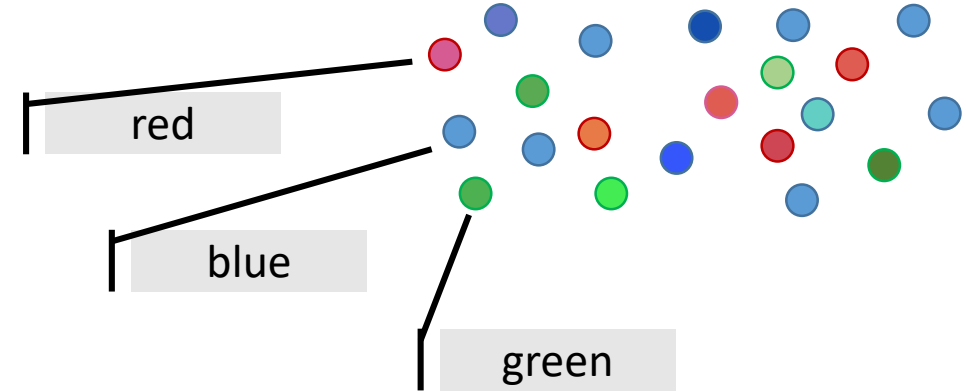
- Basic concepts and process
- Algorithms
- Examples (WEKA and SCIKIT-LEARN)

Unsupervised Learning



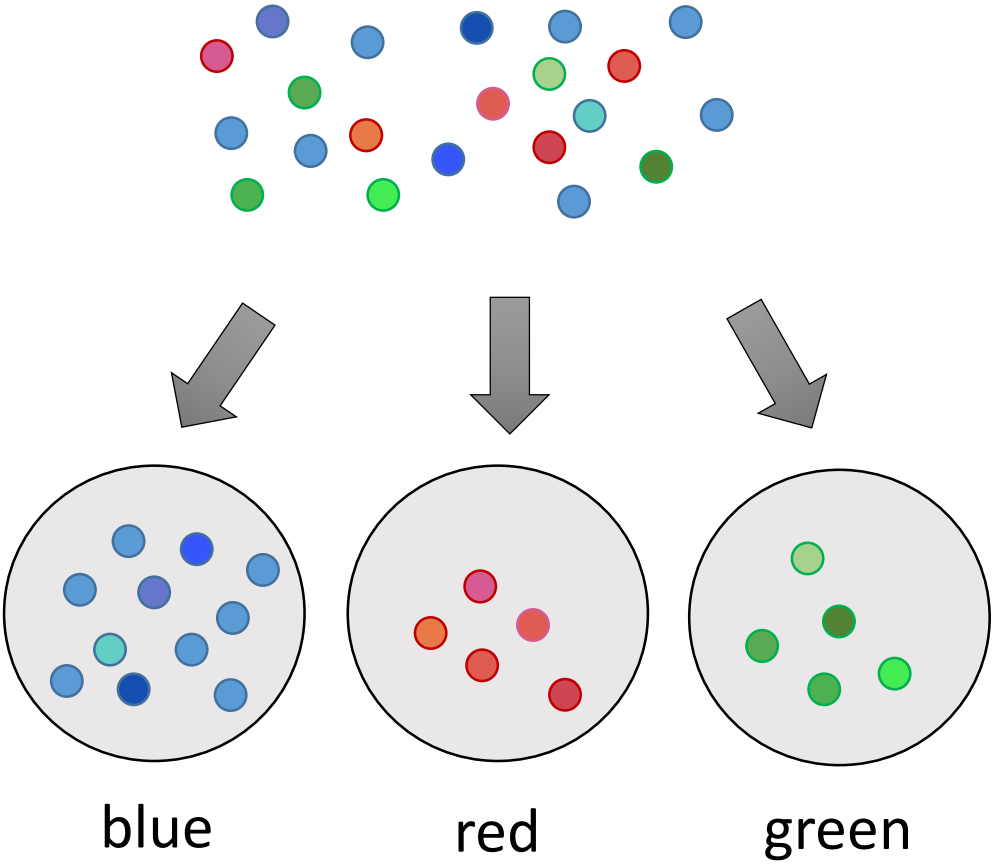
- Find patterns / clusters
- Evaluation: similarity value, classes to clusters, ...

Supervised Learning

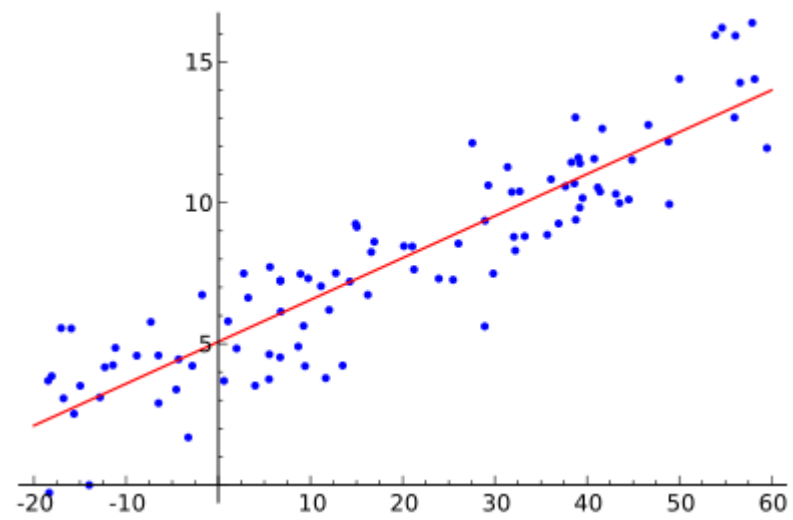


- Predict the correct label
- Evaluation: correctly classified instances, false positive rate, ...

Classification



Regression



no distinct categories, but a real value

Features / Attributes, Instances

Label /
Class attribute

	R	G	B	Color
1	227	25	59	Red
2	17	184	56	Green
3	113	125	222	Blue
4	230	67	175	Red

Features /
Attributes

Instance

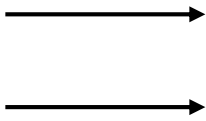
Training Set – Test Set

Dataset

5	0	4	1
9	2	1	3

Labels

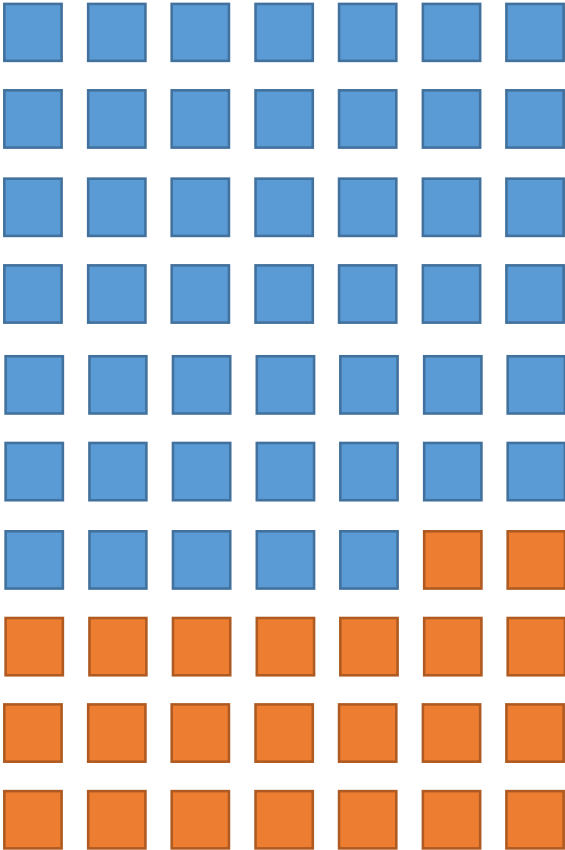
5	0	4	1
4	2	1	3



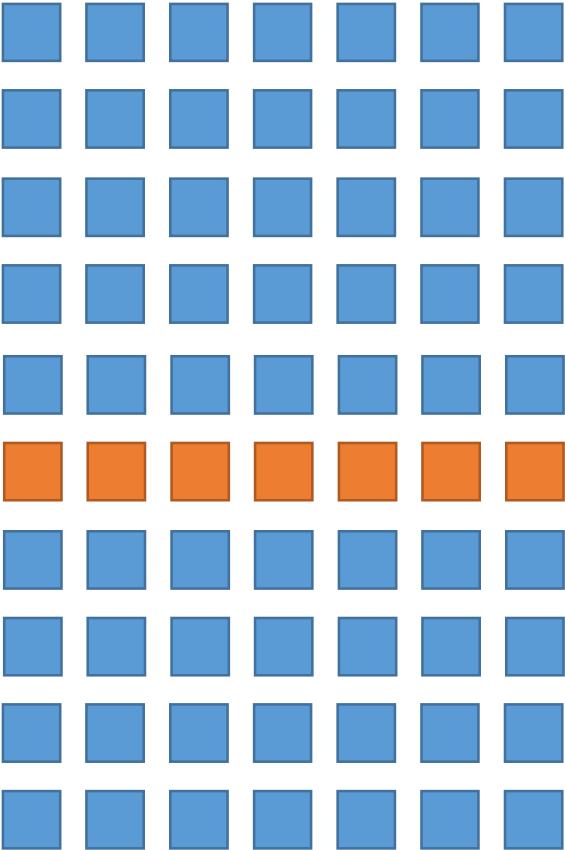
Training Set
Test Set

Validation Methods

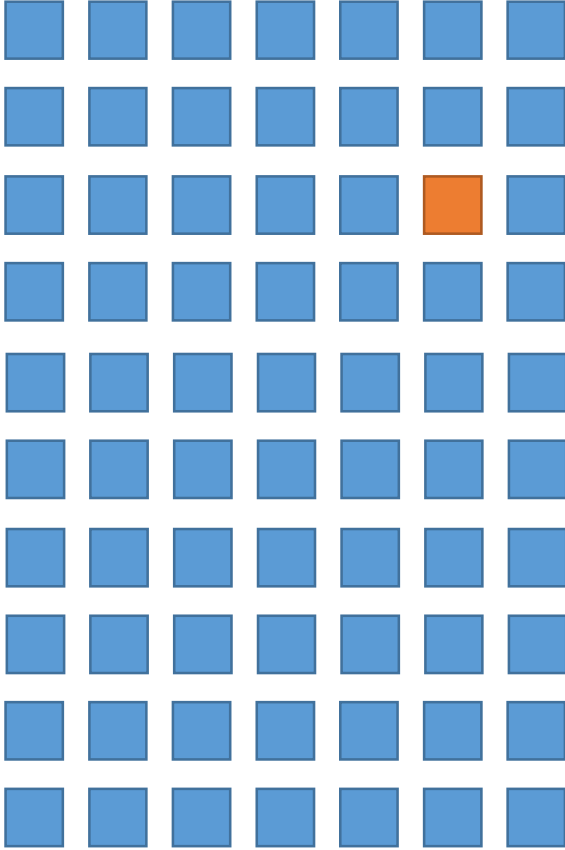
2/3 Training Set
1/3 Test Set



10-fold
Cross-validation



Leave-one-out
cross-validation



Validation Metrics

Confusion Matrix

	Classifier outcome: Positive	Classifier outcome: Negative
Condition (label): Positive	True positive	False negative
Condition (label): Negative	False positive	True negative

Accuracy: $(\Sigma \text{ True positive} + \Sigma \text{ True negative}) / \text{total}$

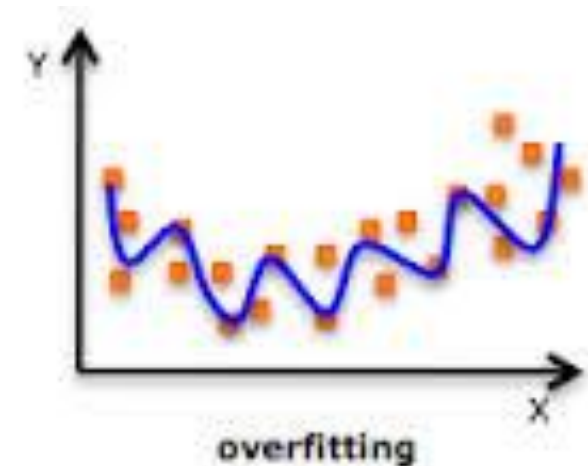
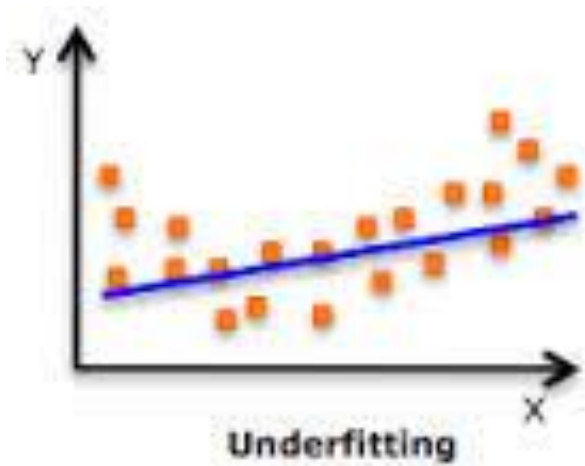
Compare to: base accuracy = percentage share of most likely category

True positive rate = Recall: $\Sigma \text{ True positive} / \Sigma \text{ condition positive}$

True negative rate: $\Sigma \text{ True negative} / \Sigma \text{ condition negative}$

Precision: $\Sigma \text{ True positive} / \Sigma \text{ Classifier outcome positive}$

Underfitting - Overfitting



Basic Process

1. Data collection
2. Feature calculation
3. Feature selection
4. Classification

Algorithms

- Naive Bayes
- Support Vector Machine
- Decision Trees

(There are many more: Neural networks, k-nearest neighbour, ...)

Naïve Bayes

- Fast and high performance
- Based on Bayes Theorem
- Assumes independence of features

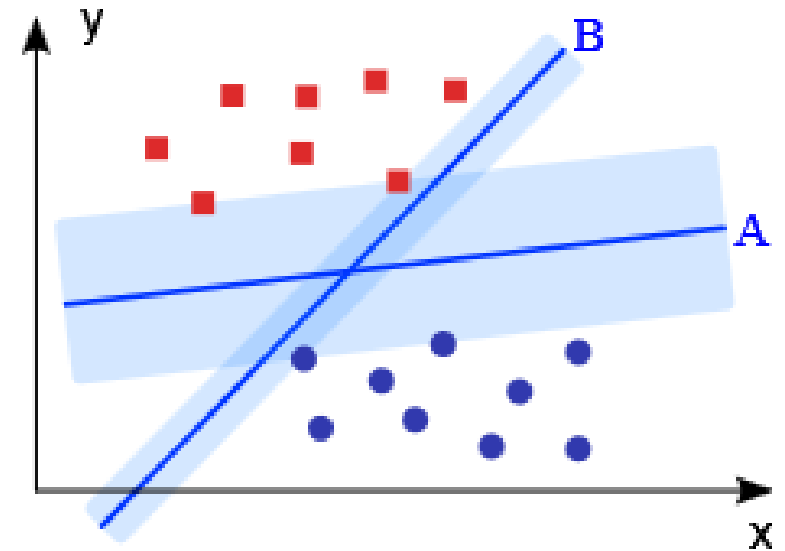
Example: e-mail classification into *spam* and *no spam*. Features: words

Bayes Theorem:
$$P(\textit{Spam}|W) = \frac{P(\textit{Spam} \cap W)}{P(W)} = \frac{P(W|\textit{Spam})P(\textit{Spam})}{P(W)}$$

$$Q = \frac{P(\textit{Spam}|W)}{P(\overline{\textit{Spam}}|W)} = \frac{P(W|\textit{Spam})P(\textit{Spam})}{P(W)} \frac{P(W)}{P(W|\overline{\textit{Spam}})P(\overline{\textit{Spam}})} = \frac{P(W|\textit{Spam})P(\textit{Spam})}{P(W|\overline{\textit{Spam}})P(\overline{\textit{Spam}})}$$

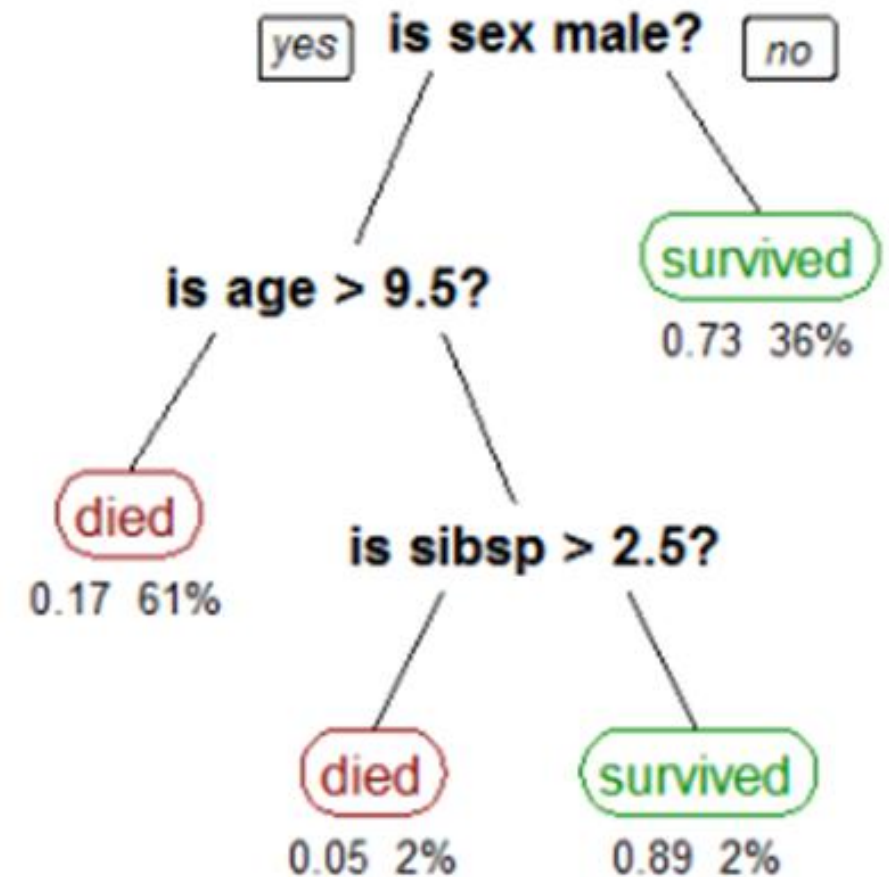
Support Vector Machine

- Divides objects in classes by maintaining a maximally large margin between the objects → *Large Margin Classifier*
- can be used for classification and regression



Decision Tree

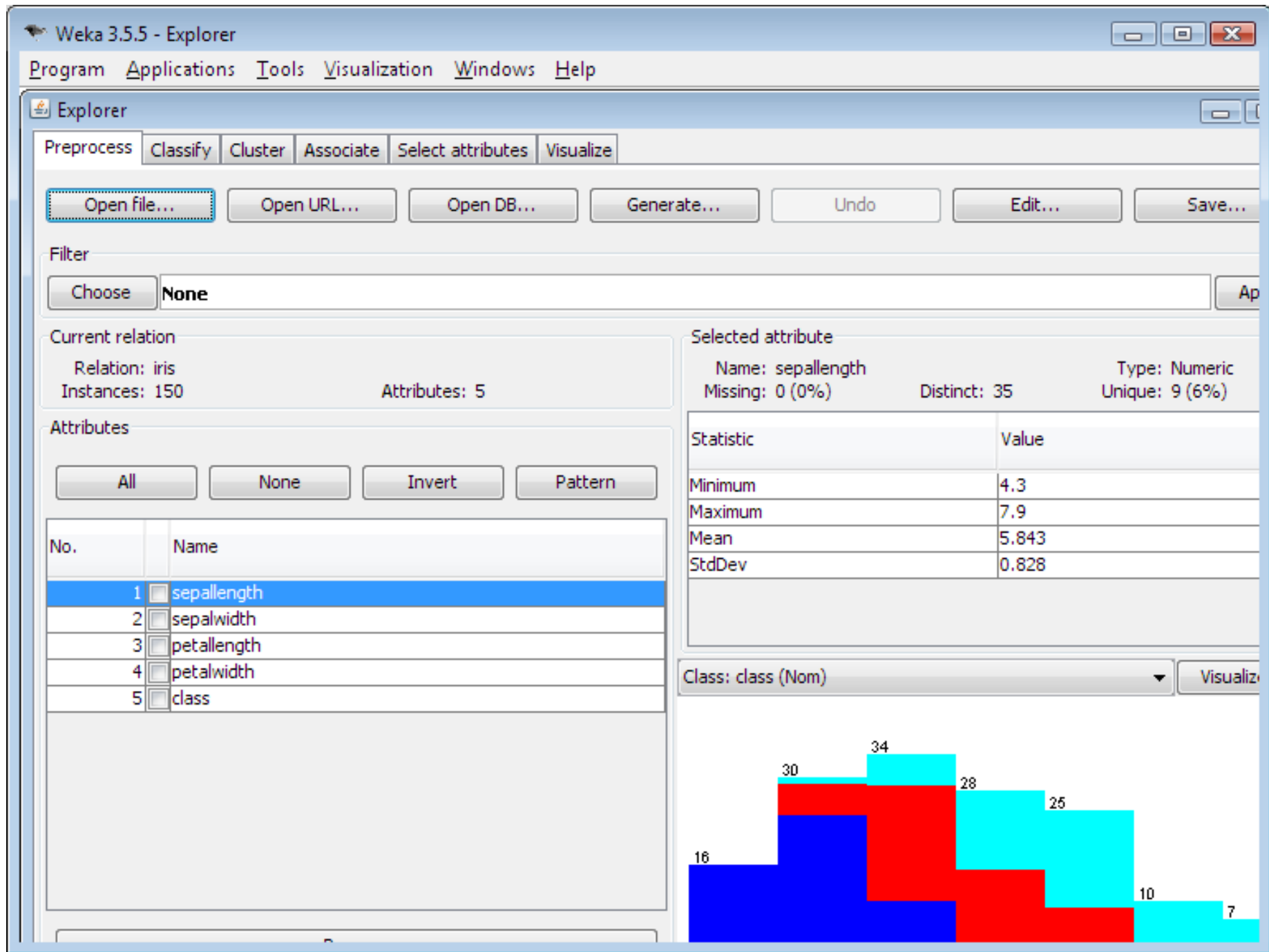
- Builds a tree to classify objects
- leaves = class labels
branches = conjunctions of features that lead to those class labels
- can be used for classification and regression



WEKA



- Java machine learning framework
- Provides a Java library and a graphical user interface
- Implements many preprocessing algorithms (filters) and classifiers
- Filters: attribute selection, transforming and combining attributes, discretization, normalization, ...
- Classifiers: Support Vector Machine (SMO), Decision Tree (J48), Naive Bayes, ...



Example Dataset: diabetes.arff

General Info:

- Number of Instances: 768
- Number of Attributes: 8 plus class
- Number of instances with label *tested_negative*: 500
- Number of instances with label *tested_positive*: 268

```
6,148,72,35,0,33.6,0.627,50,tested_positive
1,85,66,29,0,26.6,0.351,31,tested_negative
8,183,64,0,0,23.3,0.672,32,tested_positive
1,89,66,23,94,28.1,0.167,21,tested_negative
0,137,40,35,168,43.1,2.288,33,tested_positive
```

Attributes:

1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)²)
7. Diabetes pedigree function
8. Age (years)
9. Class variable (0 or 1)

Example Weka Code Part 1

```
//read data file
DataSource source = new DataSource("C:/Users/Manuela/OneDrive/Work/Teaching/HASE/diabetes.arff");
Instances data = source.getDataSet();

//set class variable
if (data.classIndex() == -1) {
    data.setClassIndex(data.attribute("class").index());
}

//Attribute selection
AttributeSelection filter = new AttributeSelection();
CfsSubsetEval eval = new CfsSubsetEval();
GreedyStepwise search = new GreedyStepwise();
search.setSearchBackwards(true);
filter.setEvaluator(eval);
filter.setSearch(search);
filter.setInputFormat(data);

// Attribute reduction
Instances filteredData = Filter.useFilter(data, filter);
```

Example Weka Code Part 2

```
for (int i = 0; i < 10; i++) {  
    int seed = i + 1;  
    Random rand = new Random(seed);  
    Instances randData = new Instances(data);  
    randData.randomize(rand);  
    if (randData.classAttribute().isNominal())  
        randData.stratify(10);  
  
    Evaluation evalJ48 = new Evaluation(randData);  
    for (int n = 0; n < 10; n++) {  
        Instances train = randData.trainCV(10, n);  
        Instances test = randData.testCV(10, n);  
  
        J48 newTree = (J48) J48.makeCopy(tree);  
        newTree.buildClassifier(train);  
        evalJ48.evaluateModel(newTree, test);  
    }  
}
```

} We do a **10 times** 10-fold cross-validation

} Randomize the data

} We do a 10 times **10-fold** cross-validation

} Set training set and test set

} build and evaluate the classifier

Interpretation of Results

Base accuracy: 65.1 %

Classifier	Features	Accuracy (%)
J48	All	74.49
J48	Selected	74.38
SMO	All	76.81
SMO	Selected	76.95
Naïve Bayes	All	75.76
Naïve Bayes	Selected	77.06

Selected Features:

- Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- Body mass index
- Diabetes pedigree function (synthesis of family history concerning diabetes)
- Age

Interpretation of Results

Confusion Matrix: Naïve Bayes, selected features

	Classifier outcome: Negative	Classifier outcome: Positive
Condition (label): Negative	436.3	63.7
Condition (label): Positive	112.5	155.5

Summary

Basic concepts of
Machine Learning

Classification

Cross-Validation

Confusion Matrix

Test Set

Overfitting

Machine Learning
algorithms

Naïve Bayes

Decision Tree

Support Vector Machine

Example

```
J48 newTree = (J48) J48.makeCopy(tree);  
newTree.buildClassifier(train);  
evalJ48.evaluateModel(newTree, test);
```

Classifier	Features	Accuracy (%)
J48	All	74.49
J48	Selected	74.38
SMO	All	76.81
SMO	Selected	76.95
Naïve Bayes	All	75.76
Naïve Bayes	Selected	77.06

Further Readings / Links to Machine Learning

- Weka Download: <http://www.cs.waikato.ac.nz/ml/weka/downloading.html>
- Weka Wiki: <http://weka.wikispaces.com/>
- Scikit-Learn: <http://scikit-learn.org/stable/>
- Sample Datasets: <http://storm.cis.fordham.edu/~gweiss/data-mining/datasets.html>
- Book about Machine Learning and Weka: <http://www.cs.waikato.ac.nz/ml/weka/book.html>
- Book about Artificial Intelligence: <http://aima.cs.berkeley.edu/>

Image Sources

Title Page: <http://www.enterprisetech.com/2014/02/11/netflix-speeds-machine-learning-amazon-gpus/>

Regression: http://www.digplanet.com/wiki/Linear_regression

Handwritten Letters: <http://yann.lecun.com/exdb/mnist/>

Overfitting: <http://pingax.com/regularization-implementation-r/>

Naïve Bayes Formulas: <http://de.wikipedia.org/wiki/Bayes-Klassifikator>

Support Vector Machine:

http://de.wikipedia.org/wiki/Support_Vector_Machine

Decision Tree: http://en.wikipedia.org/wiki/Decision_tree_learning

Weka Logo: <http://www.cs.waikato.ac.nz/ml/weka/>

Weka Screenshot: <http://commons.wikimedia.org/wiki/File:Weka-3.5.5.png>