Data Mining and Knowledge Discovery

Petra Kralj Novak

November 18, 2019

http://kt.ijs.si/petra_kralj/dmkd3.html

Data Mining and Knowledge Discovery

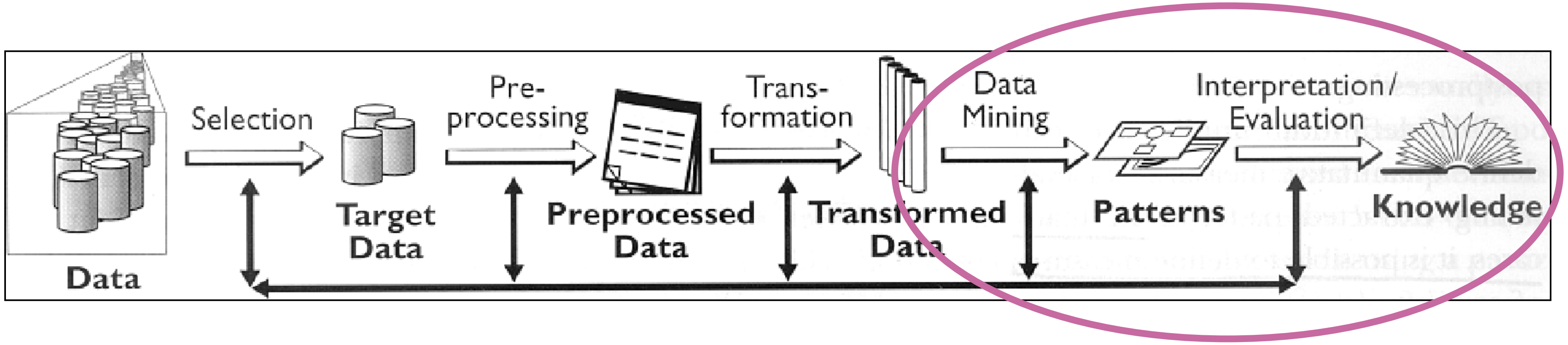
Course scope:

Prof. dr. Nada Lavrač	Introduction, rule learning, relational DM, semantic DM, embeddings
Doc. dr. Martin Žnidaršič	Ensemble methods, active learning, SVM & neural networks, correlation, causality and false patterns
Doc. dr. Petra Kralj Novak	Advanced evaluation, regression, advanced clustering Hands-on: Orange, Scikit, Keras

Course requirements:

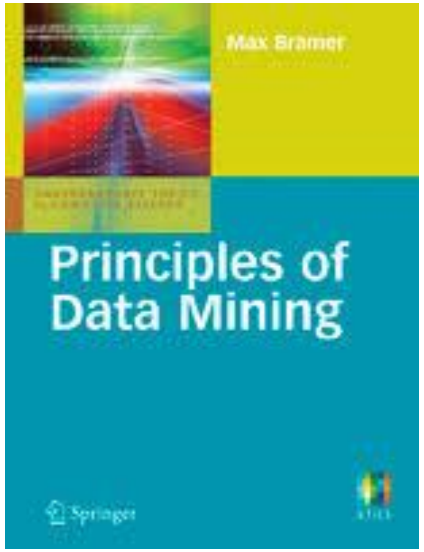
Written exam	Computational & theoretical tasks
Seminar	Analyzing your own data

Keywords

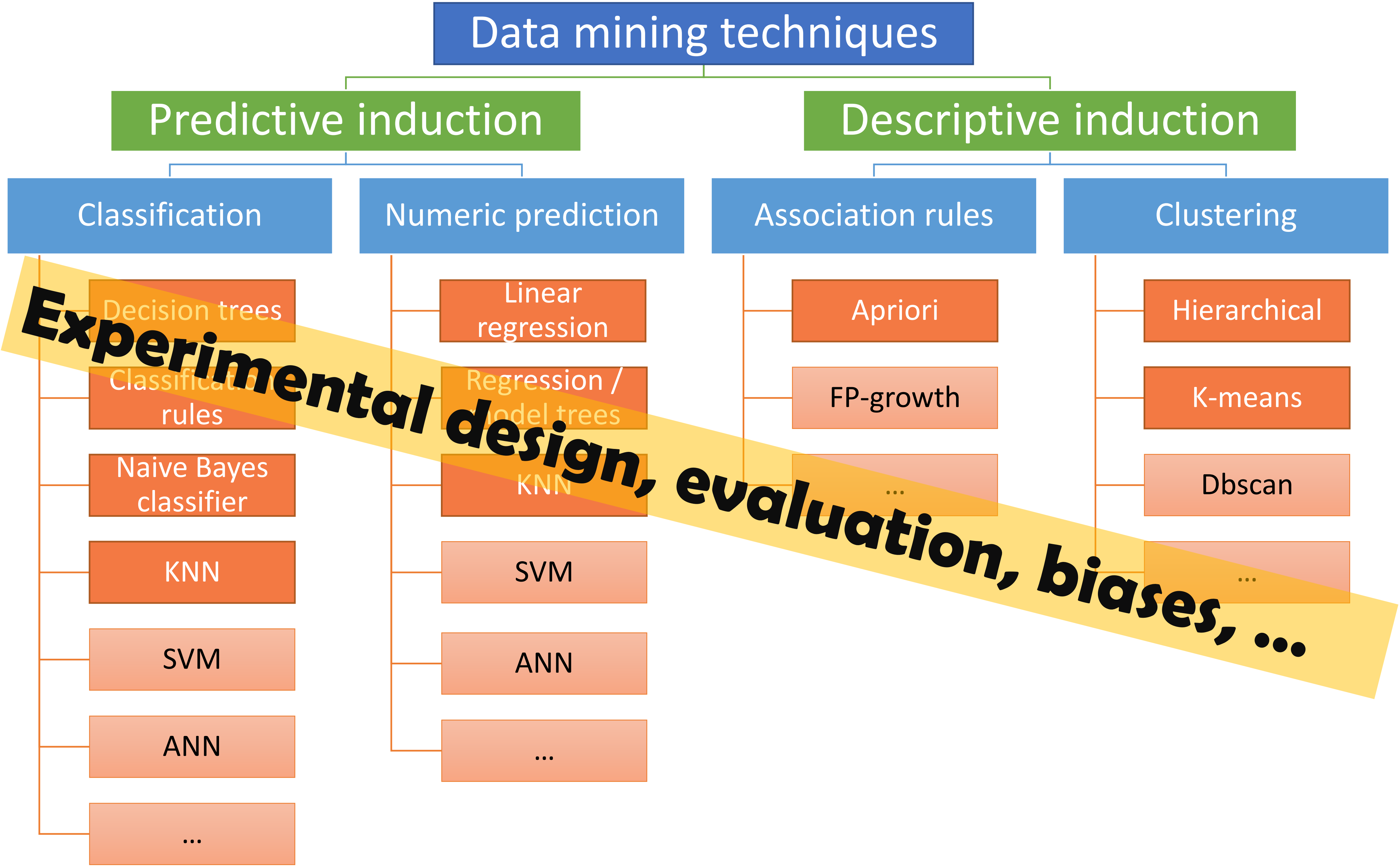


- Data
  - Attribute (feature), example (instance), attribute-value data, target variable, class, discretization, market basket data
- Algorithms
  - Decision tree induction, entropy, information gain, overfitting, Occam's razor, decision tree pruning, naïve Bayes classifier, KNN, association rules, classification rules, Laplace estimate, regression tree, model tree, hierarchical clustering, dendrogram, k-means clustering, centroid, Apriori, heuristics vs. exhaustive search, predictive vs. descriptive DM, language bias, artificial neural networks, deep learning, backpropagation,…
- Evaluation
  - Train set, test set, accuracy, confusion matrix, cross validation, true positives, false positives, ROC space, AUC, error, precision, recall, F1, MSE, RMSE, support, confidence

Bramer, Max. (2007). Principles of Data Mining. 10.1007/978-1-84628-766-4.



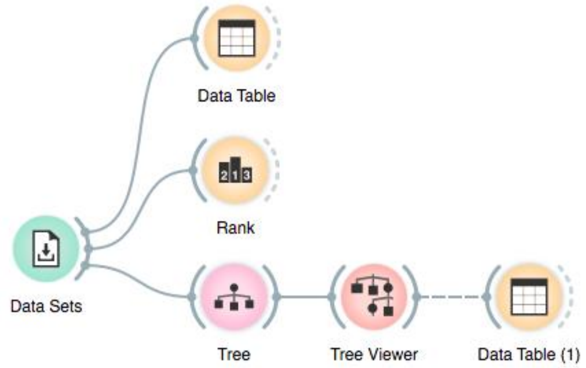
- Basic chapters about classification: 1, 2, 3, 4, 6, 8, 11
- Necessary prerequisite also for the course by prof. dr. Sašo Džeroski, doc. dr. Panče Panov: Computational Scientific Discovery from Structured, Spatial and Temporal Data


Data mining techniques
Predictive induction
Descriptive induction
Classification
Numeric prediction
Association rules
Clustering
Decision trees
Classification rules
Naive Bayes classifier
KNN
SVM
ANN
…
Linear regression
Regression / model trees
KNN
SVM
ANN
…
Apriori
FP-growth
…
Hierarchical
K-means
Dbscan
…
Experimental design, evaluation, biases, …

Hands-on

orange

- Open source machine learning and data visualization
- Interactive data analysis workflows with a large toolbox
- Visual programming
- *Demsar J, Curk T, Erjavec A, Gorup C, Hocevar T, Milutinovic M, Mozina M, Polajnar M, Toplak M, Staric A, Stajdohar M, Umek L, Zagar L, Zbontar J, Zitnik M, Zupan B (2013) Orange: Data Mining Toolbox in Python, JMLR 14(Aug): 2349–2353.*



scikit learn

- **scikit-learn i**s Gold standard of Python machine learning
- Simple and efficient tools for data mining and data analysis
- Well documented
- *Pedregosa et al. (2011) [Scikit-learn: Machine Learning in Python](), JMLR 12, pp. 2825-2830.*

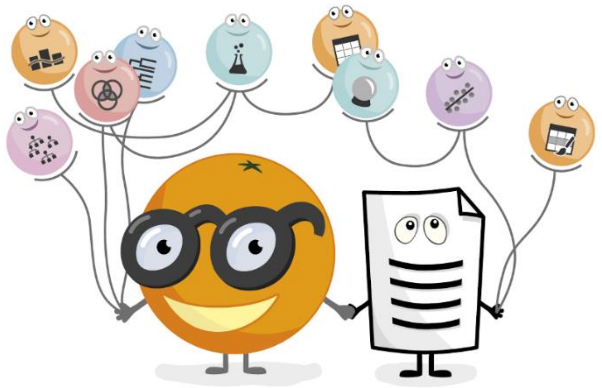
Keras

- Neural-network library written in Python.
- *Chollet, F. et al. (2015) "Keras"*

```
# ------------------------------------------------------------------------------------------
print("Train and test classification models")
classifiers = [
    # ("Naive Bayes", naive_bayes.MultinomialNB()),
    ("Logistic regression", linear_model.LogisticRegression(C=1e5, solver='lbfgs', multi_class='multinomial', max_iter=600)),
    ("MultinomialNB", MultinomialNB()),
    ("SVC", svm.LinearSVC()),
    ("SVC-RBF", svm.SVC(gamma='scale', decision_function_shape='ovo'))]


for name, classifier in classifiers:
    classifier.fit(train_data, y_train)
    predictions = classifier.predict(test_data)
    classifier.confusion_matrix = metrics.confusion_matrix(predictions, y_test, labels=["negative", "neutral", "positive"])
    classifier.accuracy = metrics.accuracy_score(predictions, y_test)
    print(name, classifier.accuracy, "\n Confusion matrix: \n", classifier.confusion_matrix)
    pickle_clf(classifier, path="./models/"+name+".pkl")
```

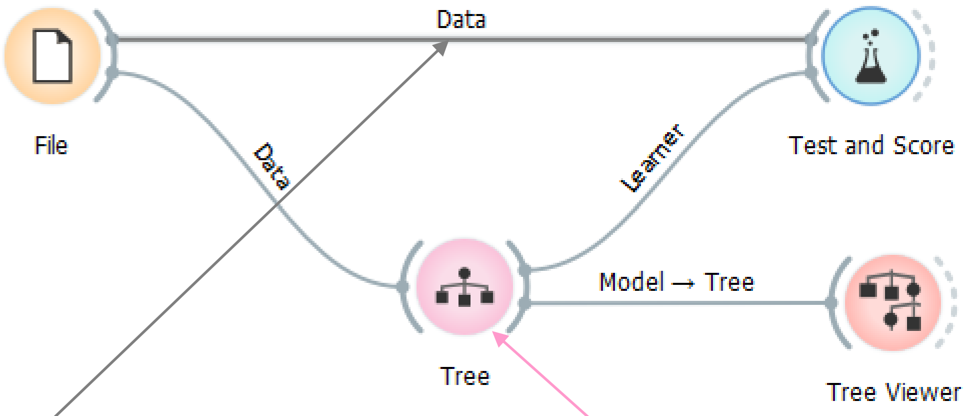
orange

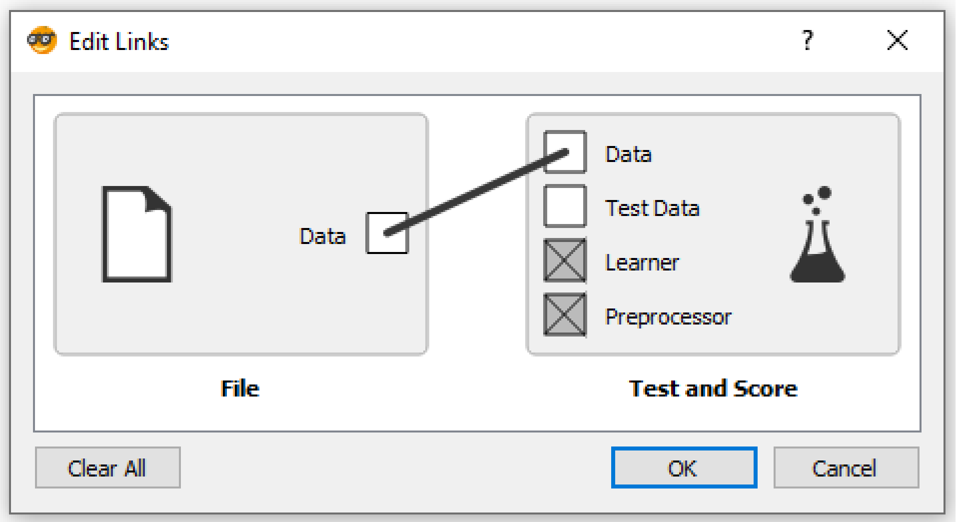


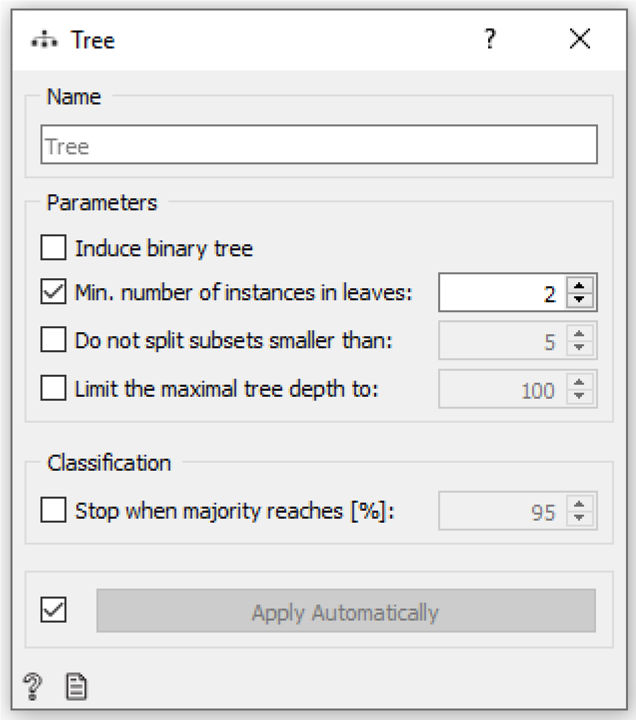
- Open source machine learning and data visualization
  - https://orange.biolab.si/
  - http://file.biolab.si/datasets/
- Interactive data analysis workflows
- Visual programming
- Based on numpy, scipy and **scikit-learn**, GUI: Qt framework

orange workflow

- Widgets: building blocks of data analysis workflows that are assembled in Orange's visual programming environment.
- A typical workflow may mix widgets for **data manipulation**, **visualization**, **modeling**, **evaluation**, …
- Widgets have inputs and outputs (typically *data objects, learner objects, classifier objects*, …) and parameters
- Interactive

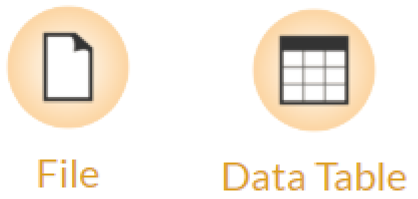






Lab exercise 1

Getting to know orange



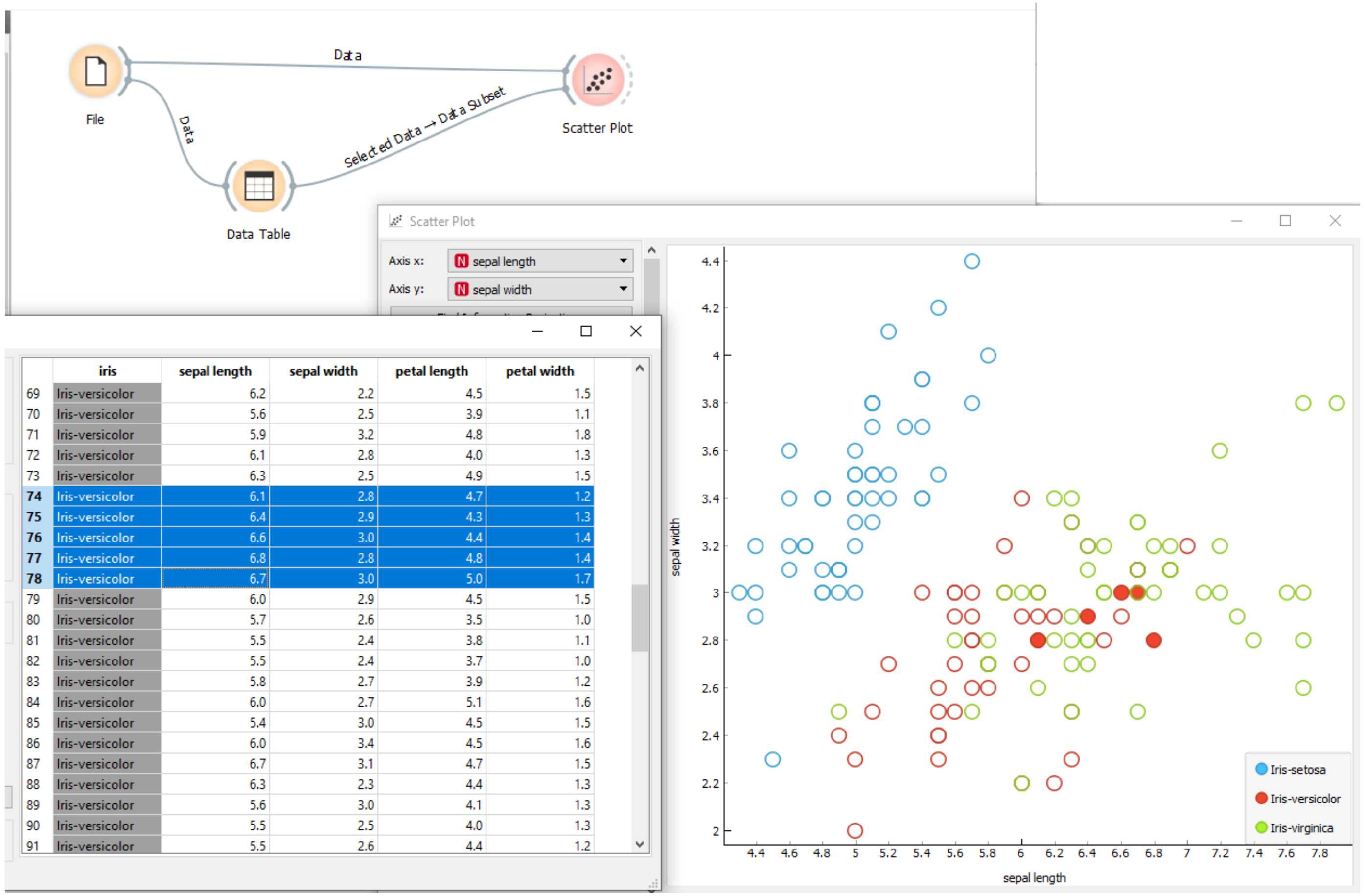
Exercise 1: Use Orange to fill in the following table

	Number of examples	Number of attributes	Number of numeric attributes	Number of categorical attributes	Target variable	Number of ordinal attributes
Zoo						
Iris						
Auto-mpg						
Wine						
Titanic						

Exercise 2: Use a text editor to view (and understand) the .tab data format.

Exercise 3: Create two interesting data visualizations with Orange.

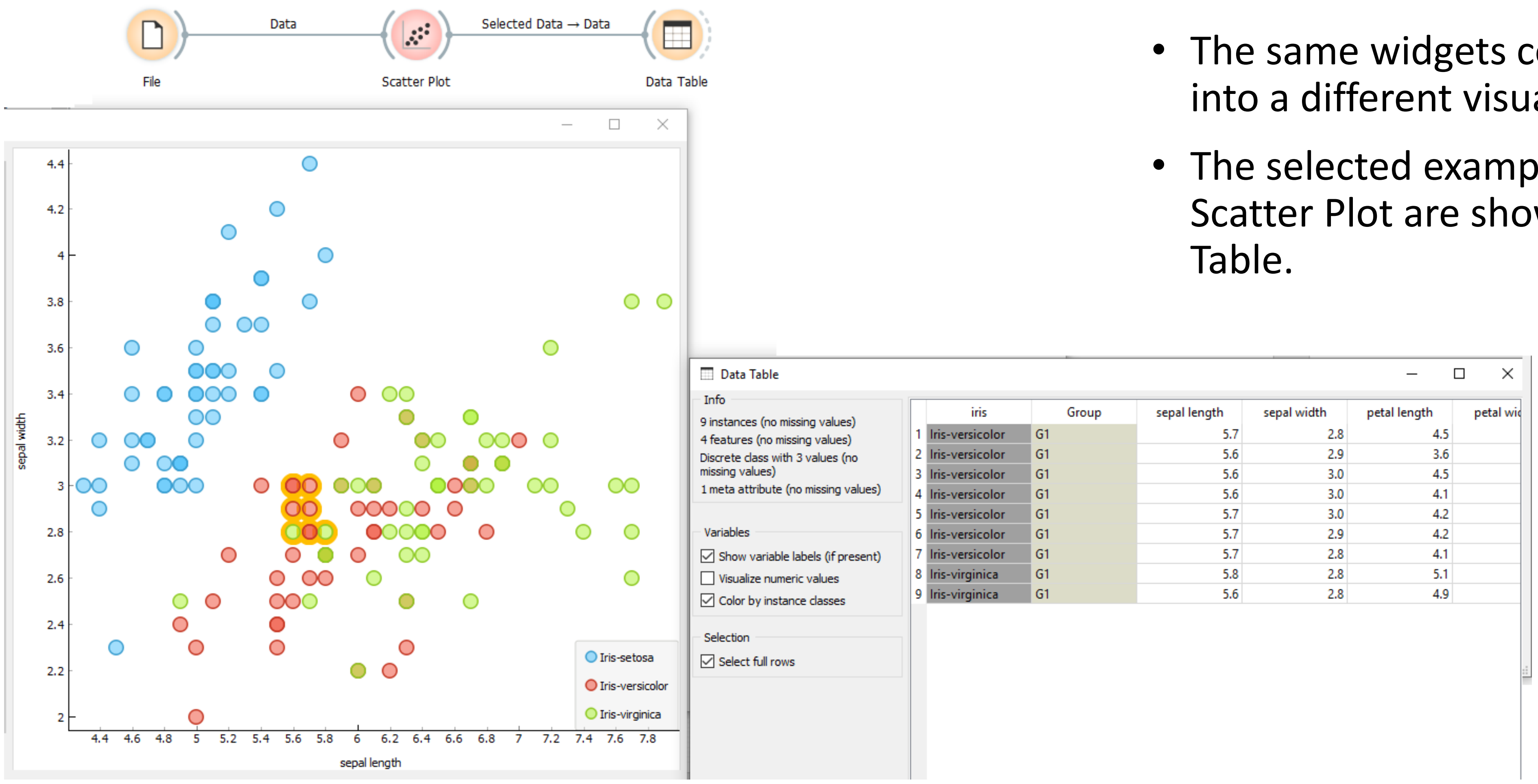
Interactive visualization in Orange



	iris	sepal length	sepal width	petal length	petal width
69	Iris-versicolor	6.2	2.2	4.5	1.5
70	Iris-versicolor	5.6	2.5	3.9	1.1
71	Iris-versicolor	5.9	3.2	4.8	1.8
72	Iris-versicolor	6.1	2.8	4.0	1.3
73	Iris-versicolor	6.3	2.5	4.9	1.5
74	Iris-versicolor	6.1	2.8	4.7	1.2
75	Iris-versicolor	6.4	2.9	4.3	1.3
76	Iris-versicolor	6.6	3.0	4.4	1.4
77	Iris-versicolor	6.8	2.8	4.8	1.4
78	Iris-versicolor	6.7	3.0	5.0	1.7
79	Iris-versicolor	6.0	2.9	4.5	1.5
80	Iris-versicolor	5.7	2.6	3.5	1.0
81	Iris-versicolor	5.5	2.4	3.8	1.1
82	Iris-versicolor	5.5	2.4	3.7	1.0
83	Iris-versicolor	5.8	2.7	3.9	1.2
84	Iris-versicolor	6.0	2.7	5.1	1.6
85	Iris-versicolor	5.4	3.0	4.5	1.5
86	Iris-versicolor	6.0	3.4	4.5	1.6
87	Iris-versicolor	6.7	3.1	4.7	1.5
88	Iris-versicolor	6.3	2.3	4.4	1.3
89	Iris-versicolor	5.6	3.0	4.1	1.3
90	Iris-versicolor	5.5	2.5	4.0	1.3
91	Iris-versicolor	5.5	2.6	4.4	1.2

- The widgets File, Data Table and Scatter Plot are connected to form a visual program.
- The selected examples in the Data Table widget are displayed as full circles in the Scatterplot.
- Note: Scatter Plot has two inputs: Data and Data subset and they need to be connected correctly.

Interactive visualization in Orange



	iris	Group	sepal length	sepal width	petal length	petal wi
1	Iris-versicolor	G1	5.7	2.8	4.5	
2	Iris-versicolor	G1	5.6	2.9	3.6	
3	Iris-versicolor	G1	5.6	3.0	4.5	
4	Iris-versicolor	G1	5.6	3.0	4.1	
5	Iris-versicolor	G1	5.7	3.0	4.2	
6	Iris-versicolor	G1	5.7	2.9	4.2	
7	Iris-versicolor	G1	5.7	2.8	4.1	
8	Iris-virginica	G1	5.8	2.8	5.1	
9	Iris-virginica	G1	5.6	2.8	4.9	

- The same widgets composed into a different visual program.
- The selected examples in Scatter Plot are shown in Data Table.

Classification

Classification in Orange

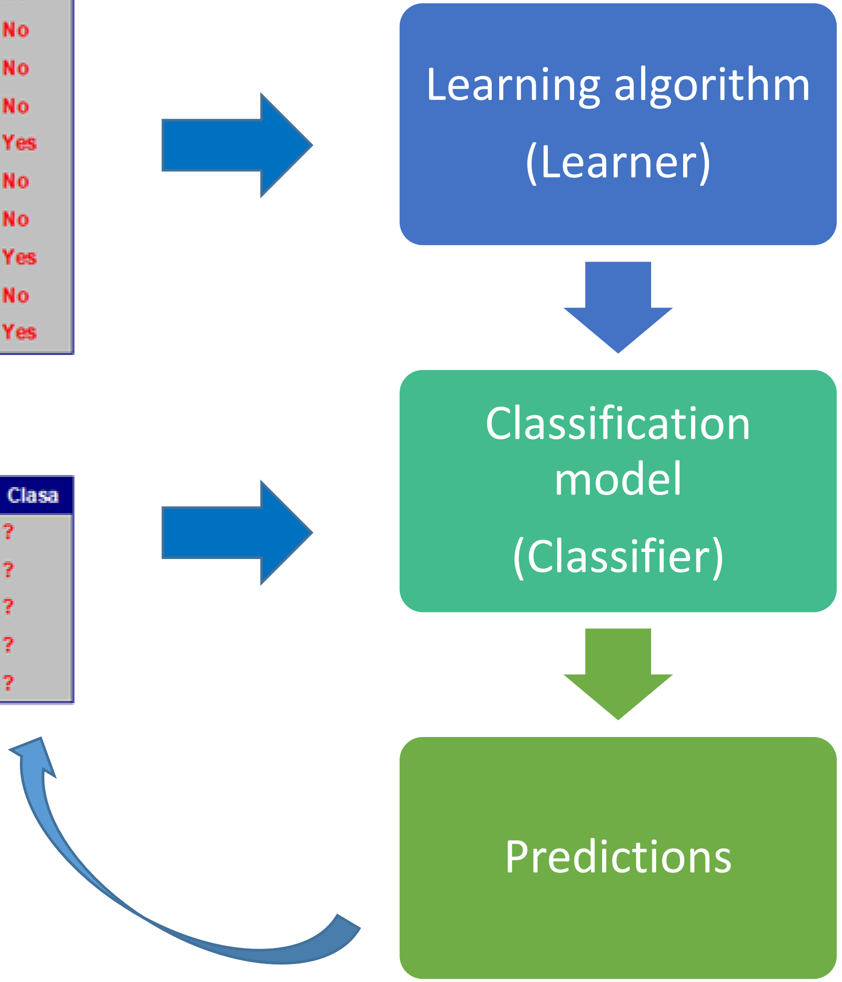
The basic classification schema

Šr	Atrib1	Atrib2	Atrib3	Clasa
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training set

Šr	Atrib1	Atrib2	Atrib3	Clasa
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

New data



- A classifier is a function that maps from the attributes to the classes
  - Classifier(attributes) = Classes
  - *f(X) = y*
- In training, the attributes and the classes are known (training examples) and we are learning a mapping function *f* (the classifier)
  - *?(X) = y*
- When predicting, both the attributes and the classifier are known, and we are assigning the classes
  - *f(X) = ?*
- What about evaluation?

The basic classification schema - evaluation

X_{train} y_{train}

Šr	Atrib1	Atrib2	Atrib3	Clasa
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training set

X_{test} y_{test} y_{pred}

Šr	Atrib1	Atrib2	Atrib3	Clasa	Clasa
11	No	Small	55K	No	No
12	Yes	Medium	80K	No	Yes
13	Yes	Large	110K	No	No
14	No	Small	95K	No	No
15	No	Large	67K	Yes	No

Testing data

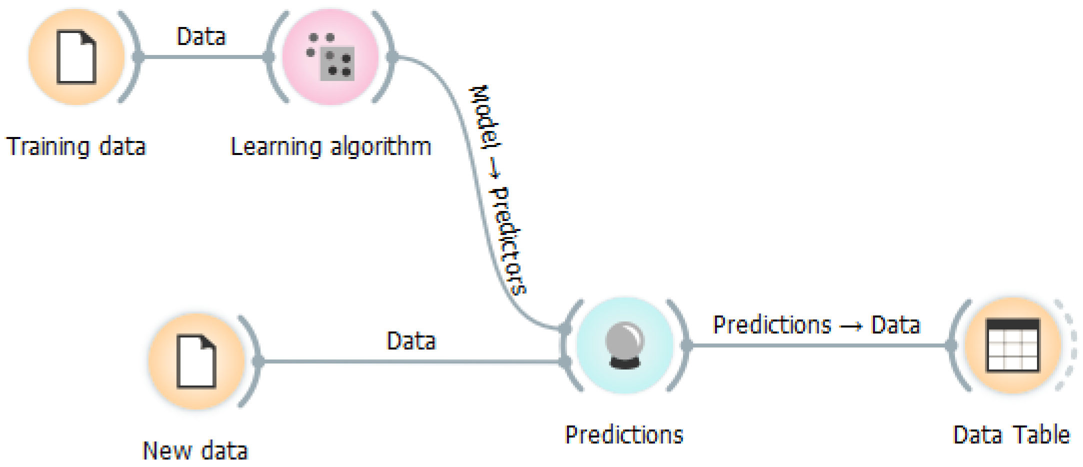
- When evaluating, *f*, *X* and y are known. We compute the predictions $y_p = f(X)$ and evaluate the difference between *Y* and Y_p.
- *Train and test data:*

  Xtrain, Xtest, ytrain, ytest

Lab exercise 2

Classification in Orange

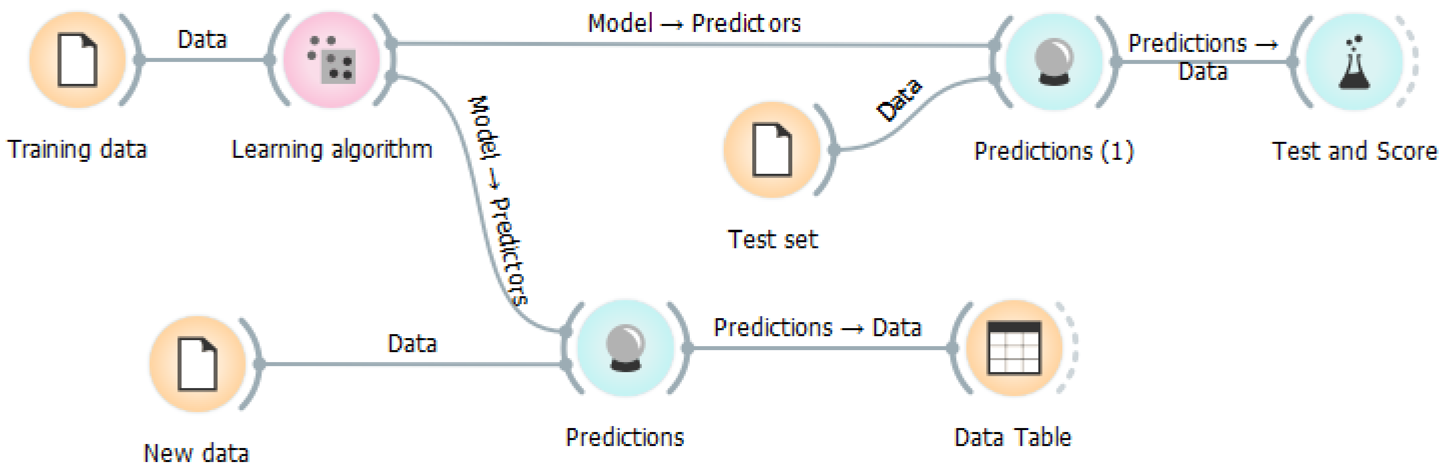
Basic classification schema in Orange



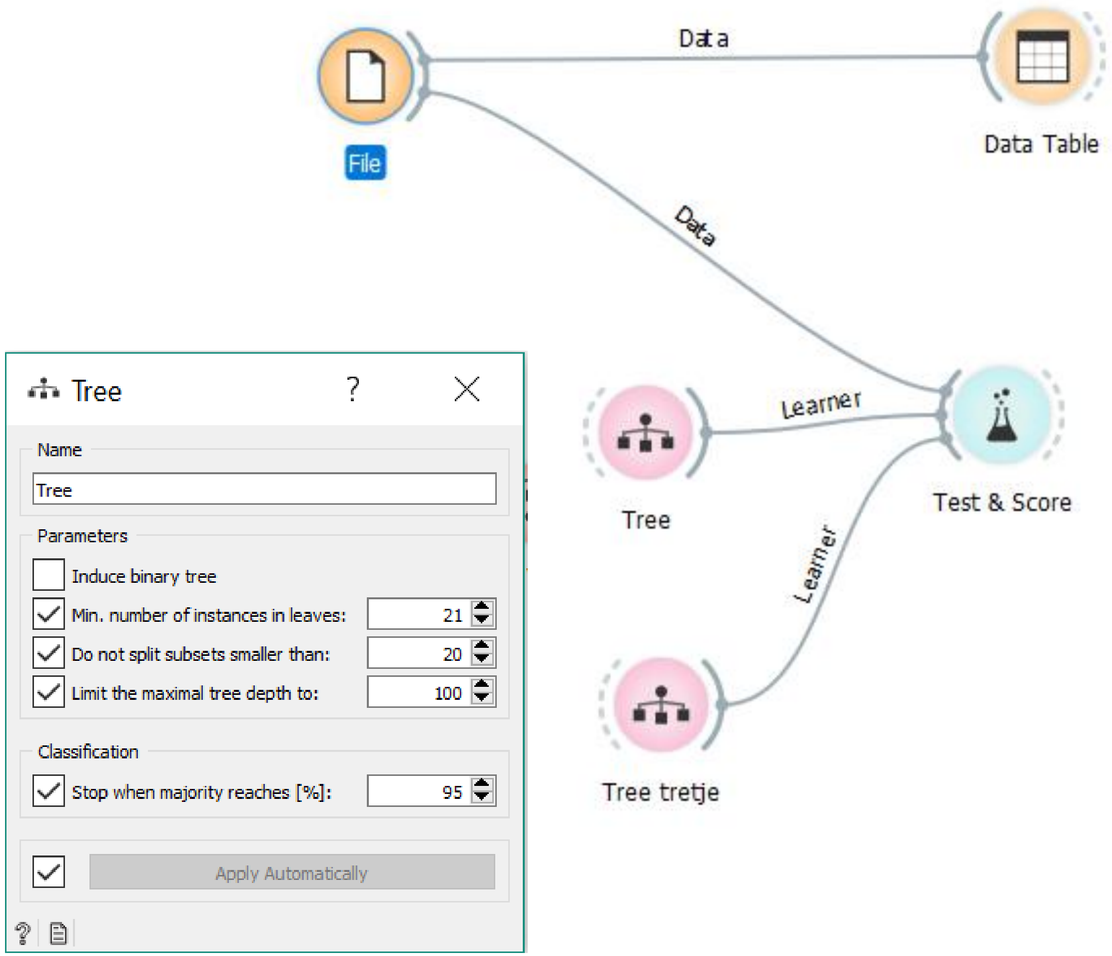
- We train the model on the train set
- We predict the target for the new instances
- There are several classification algorithms:
  - Decision trees
  - Naive Bayes classifier
  - K nearest neighbors (KNN)
  - Artificial neural networks (ANN)
  - ....

Classification with evaluation

- We train the model on the train set
- We evaluate on the test set
- We classify the new instances



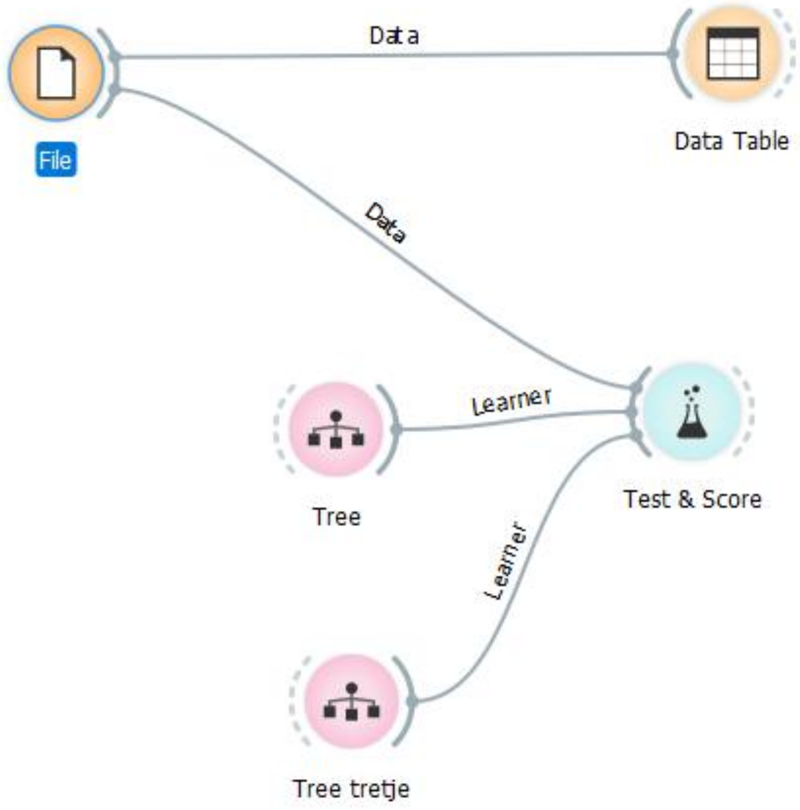
Exercise 2: Induce a decision tree



1. Dataset: “titanic”
2. Play with tree parameters

3. Repeat with the “adult” dataset
4. Evaluate tree classifiers with different parameter values

Exercise 2: Evaluate the decision tree



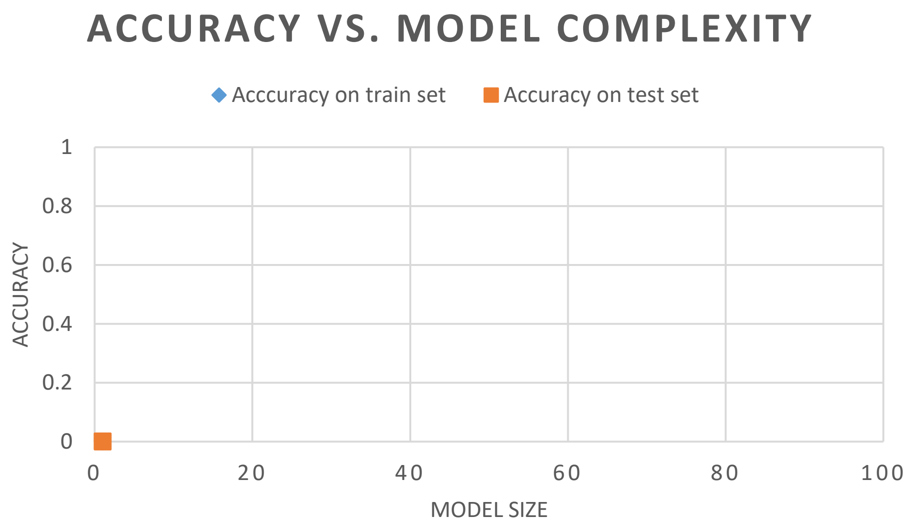
- Dataset: “zoo”
- Compare tree classifiers with different parameter values

Homework

Model complexity vs. accuracy on train and test set

Datasets:

- A-greater-then-B.csv
- Another reasonably sized classification dataset from http://file.biolab.si/datasets/



Basic classification in scikit

```
csvFileName = r".\Datasets\A-greater-then-B.csv"
df = pd.read_csv(csvFileName)

feature_cols = ['A', 'B', 'C']
target_var = 'A>B'

X = df[feature_cols].values
y = df[target_var].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)

decision_tree = tree.DecisionTreeClassifier()

decision_tree.fit(X_train, y_train)

y_pred = decision_tree.predict(X_test)

accuracy = metrics.accuracy_score(y_test, y_pred)
```

2_classification.py on http://source.ijs.si/pkraljnovak/DM_course

Refresh your memory

- Confusion matrix and ROC.
  - Bramer (2007), chapter 11: **Measuring the Performance of a Classifier**
  - Fawcett, Tom. "**An introduction to ROC analysis**." Pattern recognition letters 27.8 (2006): 861-874.