

kNN

Predict according to the nearest training instances.

Inputs

- Data: input dataset
- Preprocessor: preprocessing method(s)

Outputs

- Learner: kNN learning algorithm
- Model: trained model

The **kNN** widget uses the **kNN algorithm** that searches for k closest training examples in feature space and uses their average as prediction.

kNN

Name 1

kNN

Neighbors 2

Number of neighbors: 5

Metric: Euclidean

Weight: Uniform

3 4

Report ☒ Apply Automatically

1. A name under which it will appear in other widgets. The default name is “kNN”.
2. Set the number of nearest neighbors, the distance parameter (metric) and weights as model criteria.
 - Metric can be:
 - **Euclidean** (“straight line”, distance between two points)
 - **Manhattan** (sum of absolute differences of all attributes)
 - **Maximal** (greatest of absolute differences between attributes)
 - **Mahalanobis** (distance between point and distribution).

- The *Weights* you can use are:
 - **Uniform**: all points in each neighborhood are weighted equally.
 - **Distance**: closer neighbors of a query point have a greater influence than the neighbors further away.
- 3. Produce a report.
- 4. When you change one or more settings, you need to click *Apply*, which will put a new learner on the output and, if the training examples are given, construct a new model and output it as well. Changes can also be applied automatically by clicking the box on the left side of the *Apply* button.

Examples

The first example is a classification task on *iris* dataset. We compare the results of **k-Nearest neighbors** with the default model **Constant**, which always predicts the majority class.

Constant

Name

Constant

Report ☒ Apply Automatically

Test & Score

Sampling

☒ Cross validation

Number of folds: 10

☒ Stratified

☐ Random sampling

Repeat train/test: 10

Training set size: 66 %

☒ Stratified

☐ Leave one out

☐ Test on train data

☐ Test on test data

Target Class

(Average over classes)

Report

Evaluation Results

| Method | AUC | CA | F1 | Precision | Recall |
|----------|-------|-------|-------|-----------|--------|
| Constant | 0.500 | 0.333 | 0.167 | 0.111 | 0.333 |
| kNN | 0.989 | 0.973 | 0.973 | 0.974 | 0.973 |

The second example is a regression task. This workflow shows how to use the *Learner* output. For the purpose of this example, we used the *housing* dataset. We input the **kNN** prediction model into **Predictions** and observe the predicted values.

kNN

Name

The image displays the Orange Data Mining software interface. At the top, a workflow canvas shows a 'File' widget connected to a 'kNN' widget, which is then connected to a 'Predictions' widget. The 'kNN' widget's settings are visible on the right, showing 'Number of neighbors: 5', 'Metric: Euclidean', and 'Weight: Uniform'. Below the workflow, the 'Predictions' widget's output is shown in a table. The table has columns for 'kNN', 'MEDV', 'CRIM', 'ZN', 'INDUS', and 'CHAS'. The first 12 rows of data are displayed. On the left side of the 'Predictions' window, there is an 'Info' section showing 'Data: 506 instances.', 'Predictors: 1', and 'Task: Regression'. Below this is a 'Data View' section with a checkbox for 'Show full data set' which is checked. At the bottom left, there is an 'Output' section with checkboxes for 'Original data', 'Predictions', and 'Probabilities', all of which are checked. A 'Report' button is located at the bottom left of the 'Predictions' window.

Workflow: File → kNN → Predictions

kNN Settings:

- Neighbors
- Number of neighbors: 5
- Metric: Euclidean
- Weight: Uniform
- Report (checked)
- Apply Automatically

Predictions Table:

| | kNN | MEDV | CRIM | ZN | INDUS | CHAS |
|----|--------|--------|-------|--------|-------|-------|
| 1 | 21.780 | 24.000 | 0.006 | 18.000 | 2.310 | 0.000 |
| 2 | 22.900 | 21.600 | 0.027 | 0.000 | 7.070 | 0.000 |
| 3 | 25.360 | 34.700 | 0.027 | 0.000 | 7.070 | 0.000 |
| 4 | 26.060 | 33.400 | 0.032 | 0.000 | 2.180 | 0.000 |
| 5 | 27.100 | 36.200 | 0.069 | 0.000 | 2.180 | 0.000 |
| 6 | 27.100 | 28.700 | 0.030 | 0.000 | 2.180 | 0.000 |
| 7 | 20.880 | 22.900 | 0.088 | 12.500 | 7.870 | 0.000 |
| 8 | 19.100 | 27.100 | 0.145 | 12.500 | 7.870 | 0.000 |
| 9 | 18.400 | 16.500 | 0.211 | 12.500 | 7.870 | 0.000 |
| 10 | 19.480 | 18.900 | 0.170 | 12.500 | 7.870 | 0.000 |
| 11 | 19.280 | 15.000 | 0.225 | 12.500 | 7.870 | 0.000 |
| 12 | 22.000 | 18.900 | 0.117 | 12.500 | 7.870 | 0.000 |

Info: Data: 506 instances. Predictors: 1 Task: Regression

Data View: ☒ Show full data set

Output: ☒ Original data ☒ Predictions ☒ Probabilities

Report

| | | | | | | | |
|--------|----|--------|--------|-------|--------|-------|-------|
| report | 13 | 24.340 | 21.700 | 0.094 | 12.500 | 7.870 | 0.000 |
|--------|----|--------|--------|-------|--------|-------|-------|