

Lecture 7

Modeling with K-Nearest Neighbor

Kim, Yang Sok
Dept. of MIS, Keimyung University

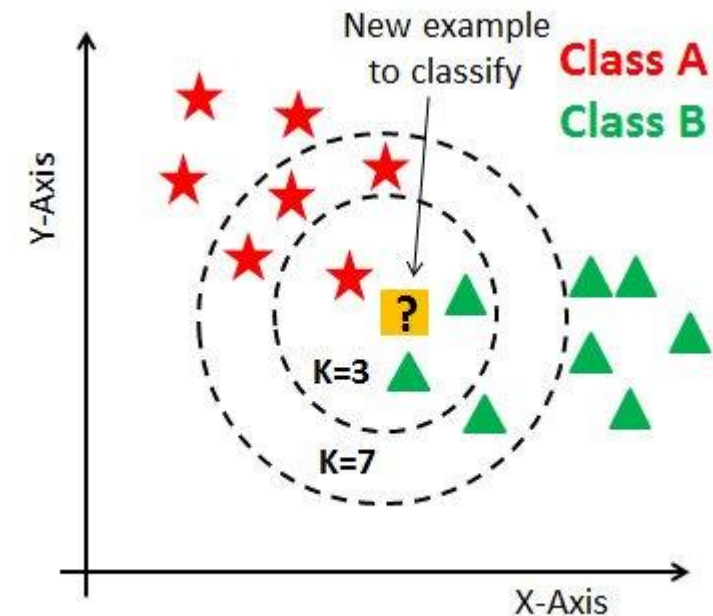
- **Introduction**
- **Algorithm**
- **Test Design**
- **Model Performance Measures**
- **Exercises**
 - Exercise 1: Modeling with split test design for K-NN (1)
 - Exercise 2: Modeling with split test design for K-NN (2)
 - Exercise 3: Modeling with x-fold cross validation design for K-NN
 - Exercise 4: Choose the Best K and Similarity for K-NN
 - Exercise 5: Explain Model Prediction
- **Conclusion**

Introduction

- KNN algorithm is one of the simplest classification algorithm and it is one of the most used learning algorithms.
- KNN is a non-parametric, lazy learning algorithm.
 - Non-parametric means that **it does not make any assumptions on the underlying data distribution.**
 - KNN is also a lazy algorithm **because it does not use the training examples to do any generalization.**
- KNN can be used for both **classification** and **regression**.

Algorithm

1. Set the number of **K** to choose the nearest neighbors
2. For each example in the data, calculate the **similarity** between the query example and the examples of the (training) dataset.
3. Pick K examples that are near to the query example
4. Return result by voting or averaging the K examples' labels



Algorithm

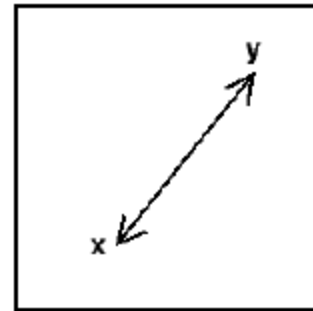
Similarity Measure

Similarity Measure

Euclidian Distance vs. Manhattan Distance vs. Minkowski Distance

• Euclidian Distance

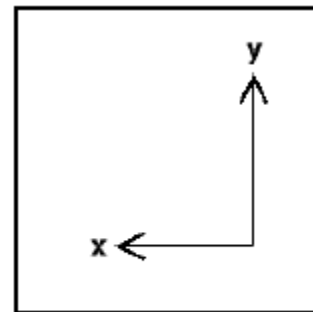
$$d(x, y) = \sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$



Euclidean

• Manhattan Distance

$$d(x, y) = \sum_{i=1}^k |x_i - y_i|$$



Manhattan

• Minkowski Distance

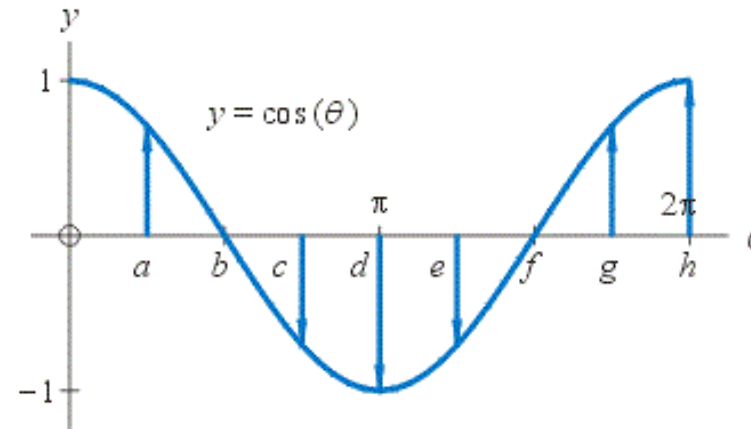
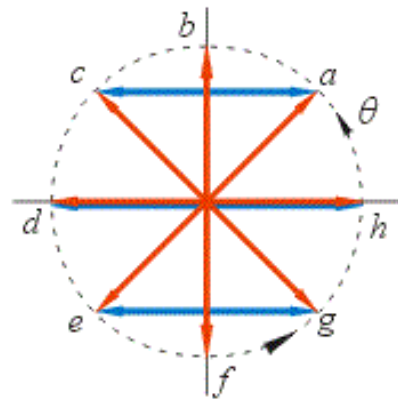
$$d(x, y) = \left(\sum_{i=1}^k |x_i - y_i|^p \right)^{1/p}$$

- $p = 1$, Manhattan Distance
- $p = 2$, Euclidean Distance
- $p = \infty$, Chebychev Distance

Similarity Measure

Cosine Similarity

- Cosine similarity measure the size of angle of two examples



- Cosine similarity is calculated by using the normalized dot product of the two attributes.

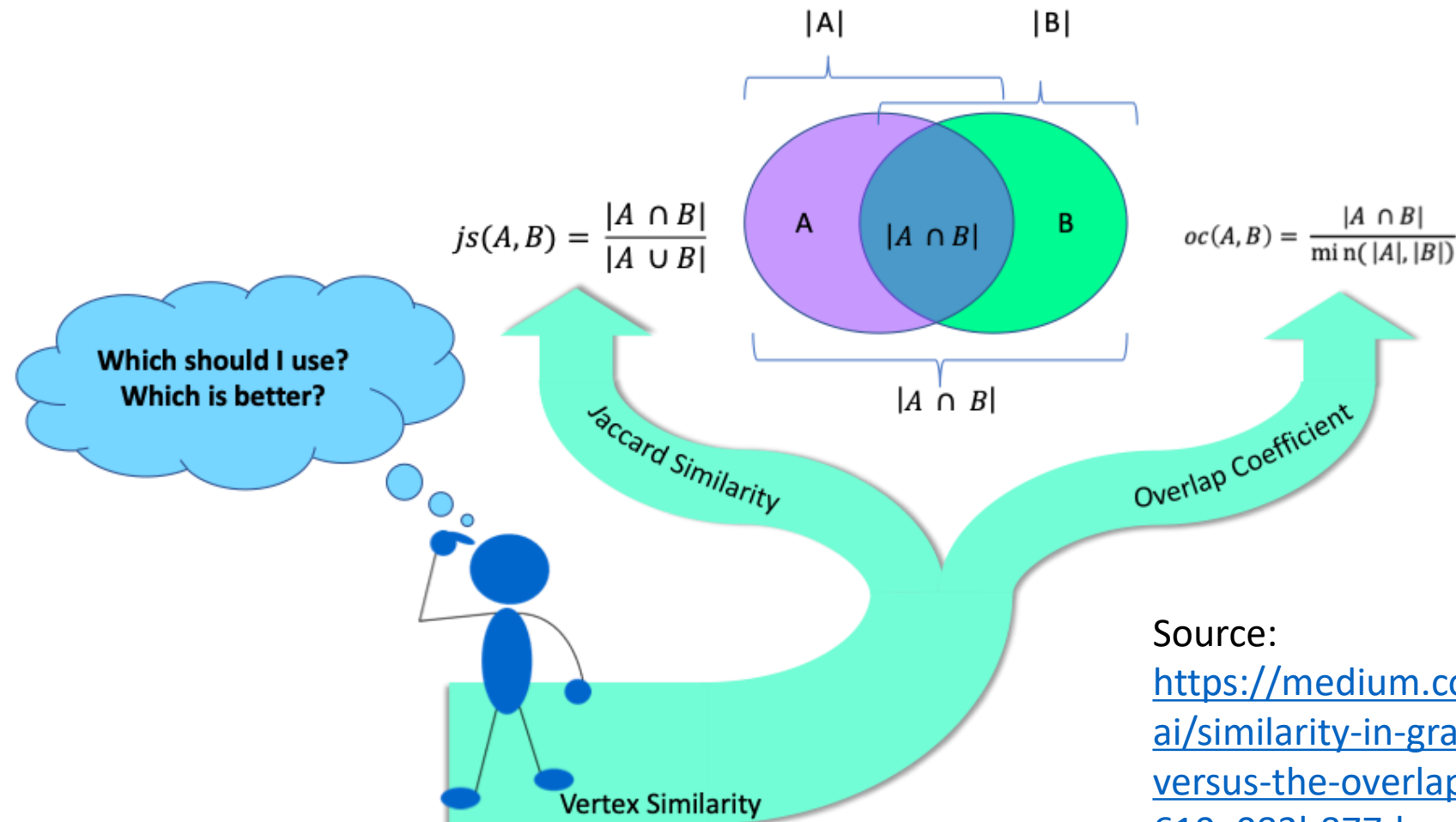
$$\text{Cosine Similarity}(x, y) = \cos(\theta) = \frac{x \cdot y}{\|x\| \cdot \|y\|} = \frac{\sum_{i=1}^k x_i \cdot y_i}{\sqrt{\sum_{i=1}^k x_i^2} \cdot \sqrt{\sum_{i=1}^k y_i^2}}$$

- Cosine similarity is particularly used in positive space, where the outcome is neatly bounded in $[0,1]$. One of the reasons for the popularity of cosine similarity is that it is very efficient to evaluate, especially for sparse vectors.

Similarity Measure

Jaccard Similarity

- Jaccard Similarity is used to find similarities between sets.



Source:

<https://medium.com/rapids-ai/similarity-in-graphs-jaccard-versus-the-overlap-coefficient-610e083b877d>

Algorithm

Choosing the Best K

Choosing Neighbors

Try & Choose

- To select the K that's right for your data, we run the KNN algorithm several times with different values of K and choose the K that reduces the number of errors we encounter while maintaining the algorithm's ability to accurately make predictions when it's given data it hasn't seen before.
- **Here are some things to keep in mind:**
 - As we decrease the value of K to 1, our predictions become less stable.
 - Inversely, as we increase the value of K , our predictions become more stable due to majority voting / averaging, and thus, more likely to make more accurate predictions (up to a certain point). Eventually, we begin to witness an increasing number of errors. It is at this point we know we have pushed the value of K too far.
 - In cases where we are taking a majority vote (e.g. picking the mode in a classification problem) among labels, we usually make K an odd number to have a tiebreaker.

Algorithm

Advantages & Disadvantages

Conclusion

Advantages & Disadvantages

- **Advantages**

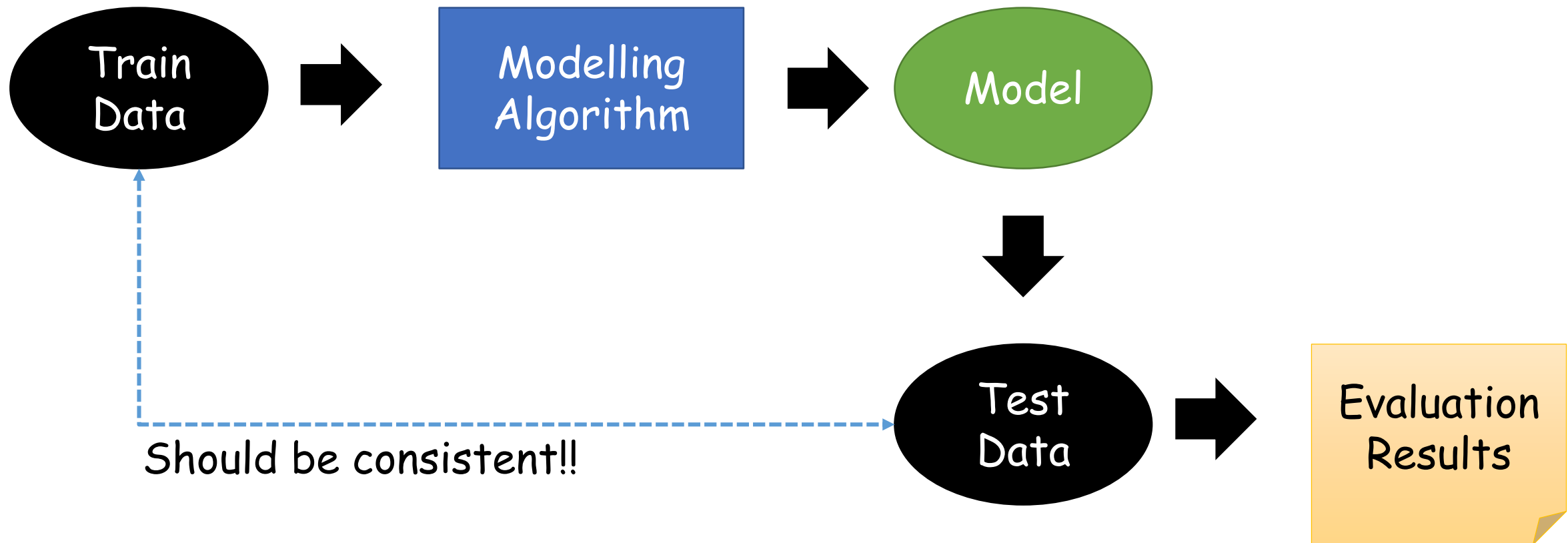
- The algorithm is simple and easy to implement.
- There's no need to build a model, tune several parameters, or make additional assumptions.
- The algorithm is versatile. It can be used for classification, regression, and search.

- **Disadvantages**

- The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase.

Test Design

- Predictive data mining algorithm learns a model for classification and prediction using a training data set and the model is evaluated against the testing dataset.

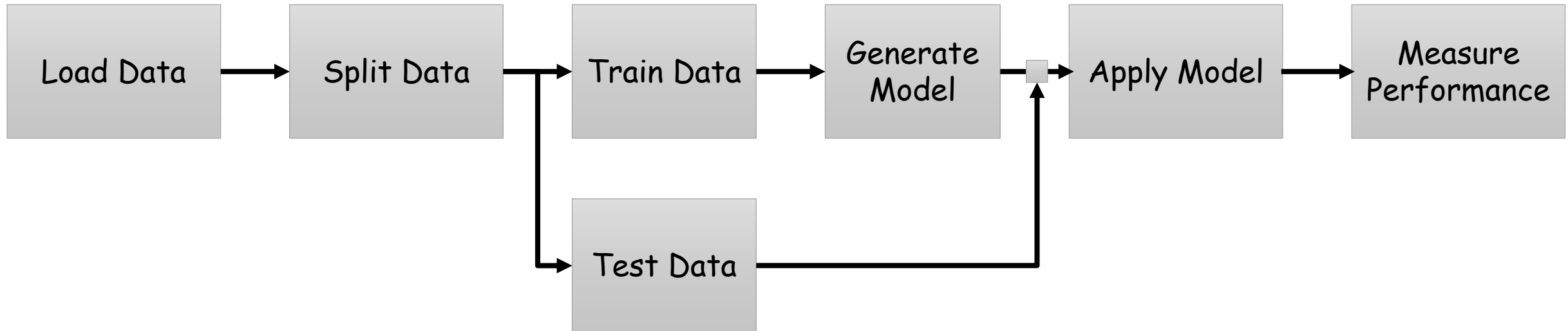


Test Design

How to generate test dataset?

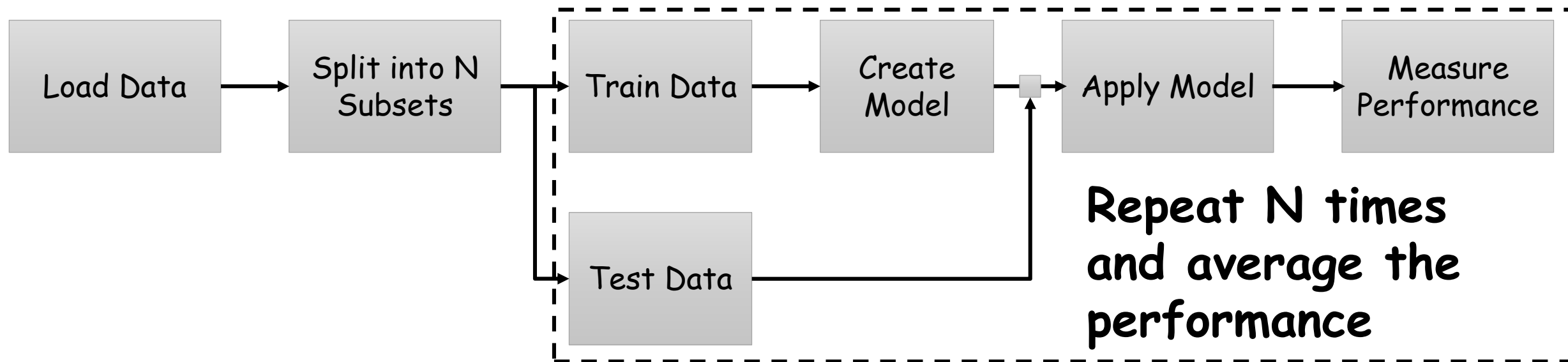
- **Split Sample Validation**

- Randomly split data into two samples (e.g., 70% = training sample, 30% = validation sample.)



- **Cross Validation**

- **Jack-knife / Leave-one-out.** The model is fitted on all the cases except one observation and is then tested on the set-aside case. This procedure can be repeated as many times as the number of observations in the original sample.
- **K-fold cross-validation.** Splits the data into K subsets; each is held out in turn as the validation set.



Exercise 1:

Modeling with split test design for K-NN (1)

Exercise 1: Modeling with split test design for K-NN

Task & Process

- **Task**
 - After loading data, generate split test design using Rapidminer
- **Process**
 - Load “red wine” dataset
 - Set “quality” as label
 - Create split validation design using <<Split Data>>
 - Create a model with the train dataset and k-NN algorithm
 - Apply the model to the test dataset
 - Set regression performance measures
 - Run the analysis process and check the performance results

Load "red wine" dataset & Set "quality" as label

The screenshot displays the RapidMiner Studio interface with a workflow in the Design view. The workflow consists of two operators: 'Read CSV' and 'Set Role'. The 'Read CSV' operator is connected to the 'Set Role' operator. The 'Set Role' operator has two output ports labeled 'exa' and 'ori'. The 'exa' port is connected to the 'Set Role' operator's 'exa' port, and the 'ori' port is connected to the 'Set Role' operator's 'ori' port. The 'Set Role' operator is highlighted with a blue circle and a callout box.

1 Load 'winequality-red.csv'

2 Add <<Set Role>>

3 Select 'quality' as 'attribute name'

4 Set 'target role' as 'label'

Parameters

- attribute name: quality
- target role: label
- set additional roles: Edit List (0)...

Operators

- Data Access (1)
 - Retrieve
- Blending (7)
- Attributes (5)
 - Names & Roles (1)
 - Set Role
 - Selection (2)
 - Select Attributes
 - Remove Useless Attributes

We found "Repository Subset Selector", "Rosette Text Analytics" and 13 more results in the Marketplace. [Show me!](#)






Help

Set Role
RapidMiner Studio Core


Tags: [Label](#), [Target](#), [Id](#), [Class](#), [Dependent](#), [Independent](#), [Special](#), [Regular](#), [Inputs](#), [Columns](#), [Attributes](#), [Features](#), [Variables](#), [Types](#), [Names & Roles](#)

Synopsis
This Operator is used to change the role

Load "red wine" dataset & Set "quality" as label





Views: Design Results Turbo Prep Auto Model


 All Studio ▼


Result History


ExampleSet (Set Role) ✕


Open in  Turbo Prep  Auto Model

Filter (1,599 / 1,599 examples): all ▼

 Data

 Statistics

 Visualizations

 Annotations

Row No.	quality	fixed acidity	volatile acidity	citric acid	residual sug...	chlorides	free sulfur d...	total sulfur d...	density	pH	sulphates	alcohol
1	5	7.400	0.700	0	1.900	0.076	11	34	0.998	3.510	0.560	9.400
2	5	7.800	0.880	0	2.600	0.098	25	67	0.997	3.200	0.680	9.800
3	5	7.800	0.760	0.040	2.300	0.092	15	54	0.997	3.260	0.650	9.800
4	6	11.200	0.280	0.560	1.900	0.075	17	60	0.998	3.160	0.580	9.800
5	5	7.400	0.700	0	1.900	0.076	11	34	0.998	3.510	0.560	9.400
6	5	7.400	0.660	0	1.800	0.075	13	40	0.998	3.510	0.560	9.400
7	5				1.600	0.069	15	59	0.996	3.300	0.460	9.400
8	7				1.200	0.065	15	21	0.995	3.390	0.470	10.000
9	7				2	0.073	9	18	0.997	3.360	0.570	9.500
10	5				6.100	0.071	17	102	0.998	3.350	0.800	10.500
11	5	6.700	0.580	0.080	1.800	0.097	15	65	0.996	3.280	0.540	9.200
12	5	7.500	0.500	0.360	6.100	0.071	17	102	0.998	3.350	0.800	10.500
13	5	5.600	0.615	0	1.600	0.089	16	59	0.994	3.580	0.520	9.900
14	5	7.800	0.610	0.290	1.600	0.114	9	29	0.997	3.260	1.560	9.100
15	5	8.900	0.620	0.180	3.800	0.176	52	145	0.999	3.160	0.880	9.200
16	5	8.900	0.620	0.190	3.900	0.170	51	148	0.999	3.170	0.930	9.200

ExampleSet (1,599 examples, 1 special attribute, 11 regular attributes)

'quality' is set as 'label'

Create split validation design using <<Split Data>>

Views: Design Results Turbo Prep Auto Model

Repository

- Import Data
- Community Samples (connected)
- Keras Samples
- DB (Legacy)
- Local Repository (admin)
 - Connections (admin)

Operators

- Split
- Blending (2)
 - Examples (1)
 - Sampling (1)
 - Split Data
- Values (1)
 - Split
- Validation (3)
 - Cross Validation
 - Split Validation

Process

Process

inp

Read CSV

Set Role

Split Data

res

res

Parameters

Split Data

partitions

sampling type automatic

use local random seed

Edit Parameter List: partitions

Edit Parameter List: **partitions**
The partitions that should be created.

ratio

0.7

0.3

Add Entry Remove Entry OK Cancel

Split Data

RapidMiner Studio Core

gs: Divide, Separate, Part, Training,
sting, Samples, Subsets, Partitions,
ampling

nopsis

is operator produces the desired
number of subsets of the given

Annotations:

- 1 Add <<Split Data>>
- 2 Click 'Edit Enumeration' button.
- 2 Select 'quality' as 'attribute name'

Create a model with the train dataset and k-NN algorithm

The image shows the RapidMiner Studio interface with a workflow diagram and several panels. The workflow consists of the following steps:

- Read CSV**: Connects to a data source.
- Set Role**: Assigns roles to the data.
- Split Data**: Divides the data into training and testing sets.
- k-NN**: The main modeling operator, which is highlighted with a blue box and an annotation.

Annotations and their corresponding actions:

- 1 Add <<k-NN>>**: Points to the **k-NN** operator in the **Operators** panel.
- 2 Set 'k' as '5'**: Points to the **k** parameter in the **Parameters** panel, which is set to 5.
- 3 Set 'measure types' as 'Numerical Measures'**: Points to the **measure types** parameter in the **Parameters** panel, which is set to **NumericalMeas...**.
- 4 Set 'numerical Measure' as 'Euclidean Distance'**: Points to the **numerical measure** parameter in the **Parameters** panel, which is set to **EuclideanDista...**.

The **Parameters** panel for the **k-NN** operator shows the following settings:

- k**: 5
- weighted vote**: ☒
- measure types**: NumericalMeas...
- numerical measure**: EuclideanDista...

The **Help** panel for the **k-NN** operator provides additional information:

- Tags**: Supervised, Classification, Regression, Neighbors, Neighbours, Knn, Instance-based, Similarity, Similarities, Euclidean, Distances, ...
- Synopsis**: This Operator generates a k-Nearest Neighbor model. It is used for classification or regression.

Apply the model to the test dataset

The screenshot displays the RapidMiner Studio interface with a workflow diagram in the center. The workflow consists of the following operators: **Read CSV** (purple), **Set Role** (pink), **Split Data** (pink), **k-NN** (green), **Multiply** (orange), and **Apply Model** (orange). The **Apply Model** operator is highlighted with a blue arrow from a callout box. The **Multiply** operator has a callout box explaining its function. The **Repository** panel on the left shows a tree structure of data sources, with **Lecture 6 Modeling** selected. The **Operator** panel on the bottom left shows a search for **Apply Model**. The **Parameters** panel on the right shows the configuration for the **Apply Model** operator.

<<Multiply>> provide input data to multiple ports

1 Add <<Apply Model>>

This operator applies the model learned from the training dataset to test data set

Repository

- Community Samples (connected)
- Keras Samples
- DB (Legacy)
- Local Repository (admin)
 - Connections (admin)
 - 2019 (admin)
 - Business Data Mining (admin)
 - Lecture 5 Data Preparation (admin)
 - Lecture 6 Modeling (admin)

Operator

Apply

- Forecasting (1)
 - Apply Forecast
- Scoring (2)
 - Confidences (1)
 - Apply Threshold
 - Apply Model

We found "Freemarket operator" and "Shapelet" in the Marketplace. [Show me!](#)

Process

Process

inp

Read CSV

Set Role

Split Data

k-NN

Multiply

Apply Model

Parameters

Apply Model

application paramet...

Edit List (0)...

create view

Hide advanced parameters

change compatibility (9.3.001)

Apply Model

RapidMiner Studio Core

Tags: Predict, Predictions, Forecasts, Scores, Scoring, Trained, Test

Synopsis

This Operator applies a model on an ExampleSet.

[Jump to Tutorial Process](#)

Recommended Operators ⓘ

Business Data Mining ©Kim Yang Sok

25

Set regression performance measures

Repository

- Import Data
- Community Samples (connected)
- Keras Samples
- DB (Legacy)
- Local Repository (admin)
 - Connections (admin)
 - 2019 (admin)
 - Business Data Mining (admin)
 - Lecture 5 Data Preparation (admin)
 - Lecture 6 Modeling (admin)

Process

Process

inp

Read CSV

Set Role

Split Data

k-NN

Multiply

Apply Model

Performance

res

res

res

res

Parameters

Performance (Performance (Regression))

- ☐ root relative squared error
- ☒ squared error ✓
- ☒ correlation ✓
- ☒ squared correlation ✓
- ☐ prediction average
- ☐ spearman rho
- ☐ kendall tau
- [Hide advanced parameters](#)

Help

Performance (Regression)

RapidMiner Studio Core

Tags: [RMSE](#), [Errors](#), [Absolute](#), [Relative](#), [Squared](#), [Predictive](#)

Synopsis

This operator is used for performance evaluation

Annotations:

1 Add <<Performance (Regression)>>

2 Check the followings:

- root mean squared error
- absolute error
- relative error
- squared error
- correlation
- squared correlation

Add this operator because data type of label is numeric.

Recommended Operators

We found "Model Management" and "Model Visualization Extension" in the Marketplace. [Show me!](#)

Business Data Mining ©Kim Yang Sok

Set regression performance measures

The screenshot displays the Orange3 data mining software interface. At the top, there is a toolbar with icons for file operations and a 'Views' section with buttons for 'Design', 'Results', 'Turbo Prep', and 'Auto Model'. A search bar on the right contains the text 'Find data, operators...etc' and a magnifying glass icon. Below the toolbar, a tab bar shows four tabs: 'Result History', 'ExampleSet (Apply Model)', 'PerformanceVector (Performance)', and 'KNNRegression (k-NN)'. The 'KNNRegression (k-NN)' tab is selected and highlighted. On the left side, there is a sidebar with two sections: 'Description' and 'Annotations'. The 'Description' section is active, showing the title 'KNNRegression' and the text: 'Weighted 5-Nearest Neighbour model for regression. The model contains 1119 examples with 11 dimensions.' A grey callout box with a pointer to the description text contains the text: 'This shows that no special information.' A black line points from a text box to the 'KNNRegression (k-NN)' tab. The text box contains the text: 'Click 'KNNRegression (k-NN)' to see the model'. A blue circle with the number '1' is positioned to the right of the text box. The bottom of the interface features a footer with the text 'Business Data Mining ©Kim Yang Sok' on the left and a dark grey box with the number '27' on the right.

Views: Design Results Turbo Prep Auto Model

Find data, operators...etc All Studio ▼

Result History ExampleSet (Apply Model) PerformanceVector (Performance) **KNNRegression (k-NN)**

KNNRegression

Description

Weighted 5-Nearest Neighbour model for regression.
The model contains 1119 examples with 11 dimensions.

Annotations

This shows that no special information.

Click 'KNNRegression (k-NN)' to see the model

1

Business Data Mining ©Kim Yang Sok

27

Run the analysis process and check the performance results

Views:

Design

Results

Turbo Prep

Auto Model

Find data, operators...etc

All Studio

Result History

ExampleSet (Apply Model)

PerformanceVector (Performance)

KNNRegression (k-NN)

Data

Statistics

Visualizations

Annotations

Open in

Turbo Prep

Auto Model

Filter (480 / 480 examples):

all

Row No.	quality	prediction(q...	fixed acidity	sug...	chlorides	free sulfur d...	total sulfur d...	density	pH	sulphur
1	5	5.774	7.400		0.076	11	34	0.998	3.510	0.560
2	5	5.406	7.800		0.098	25	67	0.997	3.200	0.680
3	5	5.774	7.400		0.076	11	34	0.998	3.510	0.560
4	5	5.388	7.400	0.660	0	1.800	0.075	13	40	0.998
5	5	5.191	7.900	0.600	0.060	1.600	0.069	15	59	0.996
6	7	5.393	7.800	0.580	0.020	2	0.073	9	18	0.997
7	5	5.373	7.500	0.500	0.360	6.100	0.071	17	102	0.998
8	5	5	8.900	0.620	0.180	3.800	0.176	52	145	0.999
9	5	5.000					0.170	51	148	0.999
10	5	5.022					0.368	16	56	0.997
11	5	5					0.106	10	37	0.997
12	5	4.800					0.084	9	67	0.997
13	6	5.189					0.085	21	40	0.997
14	6	5.993	7.800	0.645	0	2	0.082	8	16	0.996
15	5	5.169	8.300	0.655	0.120	2.300	0.083	15	113	0.997
16	5	5.196	5.200	0.320	0.250	1.800	0.103	13	50	0.996

ExampleSet (480 examples, 2 special attributes, 11 regular attributes)

Click 'ExampleSet (Apply Model)' to see the prediction results.

1

This column shows the prediction results by k-NN (k=5 and similarity measure='Euclidean'.)

Business Data Mining ©Kim Yang Sok

28

Run the analysis process and check the performance results

The screenshot shows the Orange3 software interface. At the top, there is a toolbar with icons for file operations and a 'Views' section with buttons for 'Design', 'Results', 'Turbo Prep', and 'Auto Model'. A search bar on the right contains the text 'Find data, operators...etc' and a dropdown menu labeled 'All Studio'. Below the toolbar, a tab bar shows four tabs: 'Result History', 'ExampleSet (Apply Model)', 'PerformanceVector (Performance)', and 'KNNRegression (k-NN)'. The 'PerformanceVector (Performance)' tab is active, displaying a list of performance metrics. On the left side, a sidebar contains three icons: a percentage sign for 'Performance', a document for 'Description', and a list for 'Annotations'. Two callout boxes with numbered blue circles provide instructions: Box 1 points to the 'PerformanceVector (Performance)' tab, and Box 2 points to the 'Description' icon in the sidebar. A green callout box at the bottom right contains a concluding statement.

Views: Design Results Turbo Prep Auto Model

Find data, operators...etc All Studio ▼

Result History ExampleSet (Apply Model) PerformanceVector (Performance) KNNRegression (k-NN)

PerformanceVector

PerformanceVector:

- root_mean_squared_error: 0.701 +/- 0.000
- absolute_error: 0.538 +/- 0.450
- relative_error: 9.75% +/- 9.18%
- squared_error: 0.492 +/- 0.839
- correlation: 0.456
- squared_correlation: 0.208

Performance

Description

Annotations

Click 'Performance Vector (Performance)' to see the performance results. 1

Click 'Description' to see all performance results. 2

Now you finish one cycle of analysis. You can try different k values and similarity measure to compare the results.

Exercise 2:

Modeling with split test design for K-NN (2)

Exercise 2: Modeling with split test design for K-NN

Task & Process

- **Task**

- After loading data, generate split test design using Rapidminer

- **Process**

- Load “red wine” dataset
 - Set “quality” as label
 - Create split validation design using **<<Split Validation>>**
 - Create a model with the train dataset and k-NN algorithm
 - Apply the model to the test dataset
 - Set regression performance measures
 - Run the analysis process and check the performance results

Note that all process is the same as the previous exercise except this step.

Create split validation design using <<Split Validation>>

The screenshot displays the RapidMiner Studio interface with the following components and annotations:

- Repository:** Shows a tree view of data sources. A blue circle with the number '1' is next to the 'Local Repository' section.
- Process:** The central workspace shows a workflow:
 - Read CSV:** An operator that reads data from a file.
 - Set Role:** An operator that sets the role of the data.
 - Validation:** A yellow operator with a percentage icon, highlighted with a blue circle and the number '2'. A blue arrow points from the 'Performance (Regression)' operator in the Operators panel to this 'Validation' operator.
- Parameters:** A panel on the right showing the configuration for the 'Validation (Split Validation)' operator:
 - split:** Set to 'relative'.
 - split ratio:** Set to '0.7'.
 - sampling type:** Set to 'shuffled sampling'.
 - use local random seed:** Checked.
- Operators:** A panel on the left showing a list of operators. The 'Performance (Regression)' operator is highlighted with a blue circle and the number '1'.
- Help:** A panel on the bottom right showing the documentation for the 'Split Validation' operator.

Annotations and callouts:

- Callout 1:** 'Add <<Split Validation>>' points to the 'Performance (Regression)' operator in the Operators panel.
- Callout 2:** 'Set 'split' as 'relative', 'split ratio' as '0.7' and 'sampling type' as 'shuffled sampling' points to the 'Validation' operator in the Process panel.
- Callout 3:** 'Double click this operator to get into sub process' points to the 'Validation' operator in the Process panel.

At the bottom of the interface, the text 'Business Data Mining ©Kim Yang Sok' is visible.

Create a model with the train dataset and k-NN algorithm

The screenshot displays the RapidMiner Studio interface with three numbered annotations:

- 1** Add <<k-NN>>: An arrow points from the **Operators** panel (under **Modeling** > **Predictive** > **Lazy**) to the **k-NN** operator in the **Process** canvas.
- 2** Set 'k' as '5': An arrow points from the **Parameters** panel to the **k** parameter field, which is set to 5.
- 3** Set 'measure types' as 'Numerical Measures' and 'numerical measure' as 'Euclidean Distance': An arrow points from the **Parameters** panel to the **measure types** and **numerical measure** dropdown menus.

The **Parameters** panel for **k-NN** shows the following settings:

- k**: 5
- weighted vote**: ☒
- measure types**: NumericalMeas...
- numerical measure**: EuclideanDista...

The **Help** panel for **k-NN** provides additional context:

k-NN
RapidMiner Studio Core

Tags: [Supervised](#), [Classification](#), [Regression](#), [Neighbors](#), [Neighbours](#), [Knn](#), [Instance-based](#), [Similarity](#), [Similarities](#), [Euclidean](#), [Distances](#), ...

Synopsis
This Operator generates a k-Nearest Neighbor model. It is used for classification and regression.

At the bottom, the status bar indicates: Recommended Operators ⓘ Business Data Mining ©Kim Yang Sok

Set regression performance measures

The screenshot displays the RapidMiner Studio interface with the 'Design' view selected. The process flow is as follows:

- Training:** A 'k-NN' operator receives 'tra' data and outputs 'mod' and 'exa'.
- Testing:** The 'mod' output from training is used by an 'Apply Model' operator, which also receives 'tes' data and outputs 'lab' and 'mod'.
- Performance:** A 'Performance' operator receives 'lab' and 'per' data and outputs 'ave'.

Annotations and steps:

- 1** Add <<Apply Model>>: Points to the 'Apply Model' operator in the testing section.
- 2** Add <<Performance (Regression)>>: Points to the 'Performance' operator in the testing section.
- 3** Check the followings:
 - root mean squared error
 - absolute error
 - relative error
 - squared error
 - correlation
 - squared correlation

The 'Parameters' panel on the right shows the configuration for the 'Performance (Performance (Regression))' operator:

- main criterion:** first
- Performance Measures:**
 - ☒ root mean squared error
 - ☒ absolute error ✓
 - ☒ relative error ✓
 - ☐ relative error lenient
 - ☐ relative error strict
 - ☐ normalized absolute error
- [Hide advanced parameters](#)

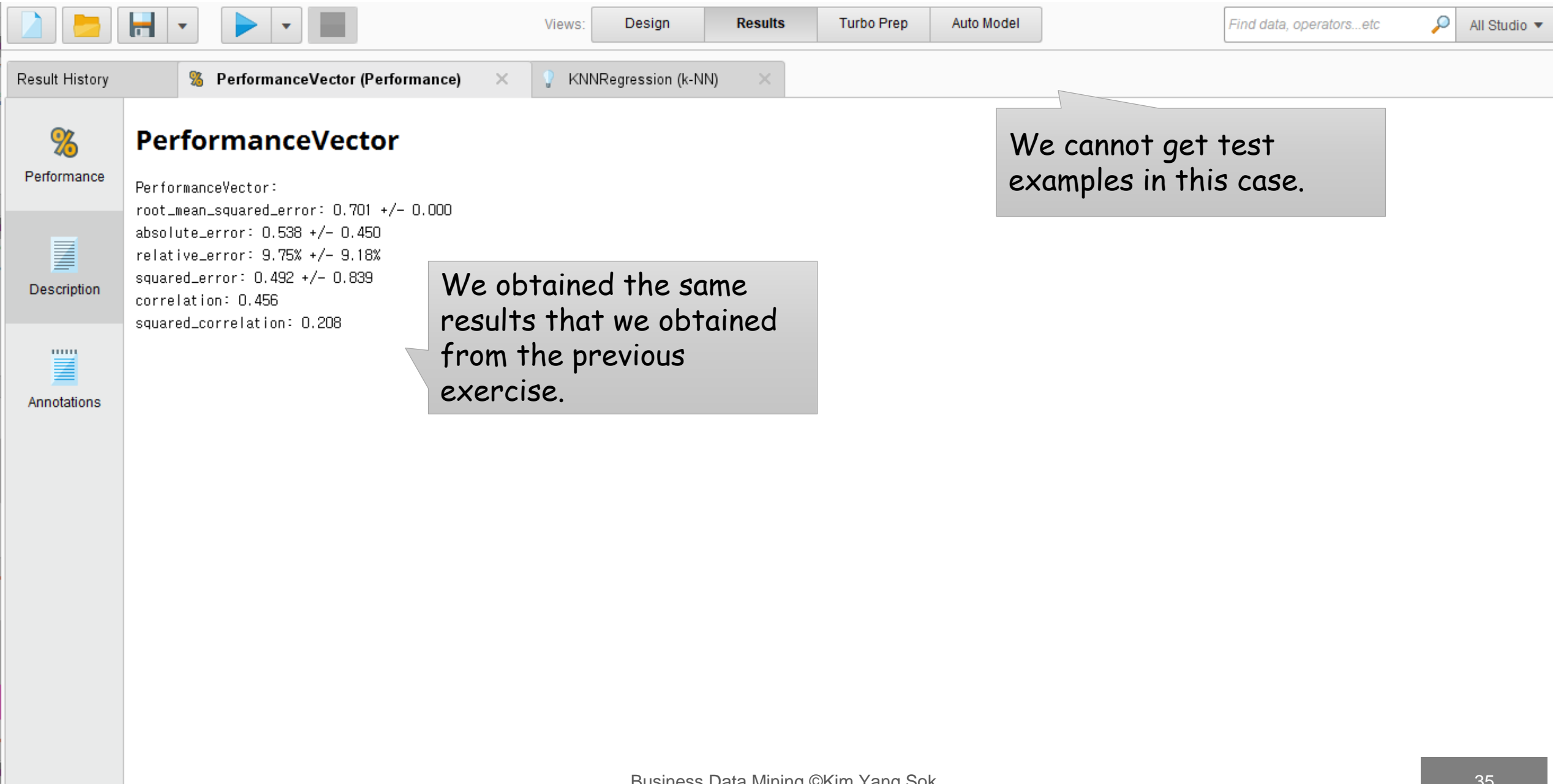
The 'Operators' panel on the left shows the search results for 'Perfo', with 'Performance (Regression)' selected under the 'Predictive (7)' category.

Help Panel:

- Performance (Regression)**
RapidMiner Studio Core
- Tags:** RMSE, Errors, Absolute, Relative, Squared, Predictive
- Synopsis:** This operator is used for statistical performance evaluation of regression task

Click to select, drag to move.

Run the analysis process and check the performance results



The screenshot shows the Orange3 data mining software interface. At the top, there is a toolbar with icons for file operations and a 'Views' section with buttons for 'Design', 'Results' (which is selected), 'Turbo Prep', and 'Auto Model'. To the right of the toolbar is a search bar with the text 'Find data, operators...etc' and a magnifying glass icon, followed by a dropdown menu labeled 'All Studio'. Below the toolbar, there is a tab bar with three tabs: 'Result History', 'PerformanceVector (Performance)' (which is active), and 'KNNRegression (k-NN)'. The main area of the 'PerformanceVector' tab is divided into three sections: 'Performance' (with a percentage icon), 'Description' (with a document icon), and 'Annotations' (with a list icon). The 'Performance' section displays the following metrics: PerformanceVector: root_mean_squared_error: 0.701 +/- 0.000, absolute_error: 0.538 +/- 0.450, relative_error: 9.75% +/- 9.18%, squared_error: 0.492 +/- 0.839, correlation: 0.456, and squared_correlation: 0.208. There are two callout boxes: one on the right side of the 'Performance' section stating 'We cannot get test examples in this case.', and another on the left side of the 'Description' section stating 'We obtained the same results that we obtained from the previous exercise.'

Views: Design Results Turbo Prep Auto Model

Find data, operators...etc All Studio

Result History PerformanceVector (Performance) KNNRegression (k-NN)

PerformanceVector

PerformanceVector:
root_mean_squared_error: 0.701 +/- 0.000
absolute_error: 0.538 +/- 0.450
relative_error: 9.75% +/- 9.18%
squared_error: 0.492 +/- 0.839
correlation: 0.456
squared_correlation: 0.208

We cannot get test examples in this case.

We obtained the same results that we obtained from the previous exercise.

Business Data Mining ©Kim Yang Sok

35

Exercise 3:

Modeling with x-fold cross validation design for K-NN

Exercise: Modeling with split test design for K-NN

Task & Process

- **Task**

- After loading data, generate **x-fold cross validation design** using Rapidminer

- **Process**

- Load “red wine” dataset
- Change the data type of “quality” into polynomial
- Set “quality” as label
- Create x-fold cross validation design using **<<Cross Validation>>**
- Create a model with the train dataset and k-NN algorithm
- Apply the model to the test dataset
- Set **classification** performance measures
- Run the analysis process and check the performance results

Change the data type of "quality" into polynomial

The screenshot displays the RapidMiner Studio interface with the following components and annotations:

- Repository:** Contains a tree view of data sources. A callout box labeled "1" points to the "Numerical to Polynomial" operator in the "Types (9)" list, with the text "Add <<Numerical to Polynomial>>".
- Process:** The central workspace showing a workflow:
 - Read CSV:** The first operator in the process.
 - Numerical to Polynomial:** The second operator, highlighted with a callout box labeled "1".
 - Set Role:** The third operator, with a callout box labeled "Set 'quality' as label".
- Parameters:** A panel on the right showing the configuration for the "Numerical to Polynomial" operator:
 - attribute filter type:** Set to "single" (callout box labeled "2" with text "Set 'attribute filter type' as 'single'").
 - attribute:** Set to "quality" (callout box labeled "3" with text "Set 'attribute' as 'quality'").
 - Options for "invert selection" and "include special attributes" are unchecked.
- Help:** A panel at the bottom right showing the documentation for the "Numerical to Polynomial" operator, including tags and a synopsis.
- Bottom Bar:** Displays "Recommended Operators" and the text "Business Data Mining ©Kim Yang Sok".

Create x-fold cross validation design using <<Cross Validation>>

Repository

- Import Data
- Community Samples (connected)
- Keras Samples
- DB (Legacy)
- Local Repository (admin)
 - Connections (admin)
 - 20
 - Lecture 6 Modeling (admin)

Operators

- Performance (17)
 - Predictive (7)
 - Performance (Classification)
 - Performance (Binominal Clas
 - Performance (Regression)
 - Performance (Costs)

Process

Process

inp

Read CSV

Numerical to Polyno...

Set Role

Cross Validation

2 Set 'number of folds' as '10'

3 Set 'sampling type' as 'stratified sampling'

4 Double click this operator to get into the sub process

Parameters

Cross Validation

- ☐ split on batch attribute
- ☐ leave one out
- number of folds ☒ 10
- sampling type ☒ stratified sampli...
- ☐ use local random seed
- ☒ enable parallel execution
- [Hide advanced parameters](#)
- ☒ [Change compatibility \(9.3.001\)](#)

Help

Cross Validation

Concurrency

Tags: [Cross-Validations](#), [Cross-validations](#), [Folds](#), [K-Folds](#), [K-folds](#), [Validations](#), [Estimations](#), [Evaluations](#), [Performances](#), [Splitting](#), [X-Validation](#), [X-Prediction](#), [Validation](#)

Synopsis

Business Data Mining ©Kim Yang Sok

39

Create a model with the train dataset and k-NN algorithm

The screenshot displays the RapidMiner Studio interface with three numbered callouts illustrating the steps to create a k-NN model:

- 1 Add <<k-NN>>**: A blue arrow points from the **Operators** panel (under **Modeling** > **Predictive** > **Lazy**) to the **k-NN** operator in the **Process** canvas.
- 2 Set 'k' as '5'**: A box points to the **Parameters** panel where the **k** value is set to 5.
- 3 Set 'measure types' as 'Numerical Measures' and 'numerical measure' as 'Cosine Similarity'**: A box points to the **Parameters** panel where **measure types** is set to **NumericalMeasures** and **numerical measure** is set to **CosineSimilarity**.

Repository

- Import Data
- Community Samples (connected)
- Keras
- Connections (admin)
- 2019 (admin)
- Business Data Mining (admin)
 - Lecture 5 Data Preparation (admin)
 - Lecture 6 Modeling (admin)

Operators

- k-NN
- Modeling (1)
 - Predictive (1)
 - Lazy (1)
 - k-NN
- Extensions (8)
 - Recommenders (8)
 - Item Recommendation (4)

Process

Process > Cross Validation

Training

Testing

Parameters

- k-NN
- k ✓ 5
- ☒ weighted vote ✓
- measure types NumericalMeasures
- numerical measure CosineSimilarity

Help

k-NN
RapidMiner Studio Core

Tags: Supervised, Classification, Regression, Neighbors, Neighbours, Knn, Instance-based, Similarity, Similarities, Euclidean, Distances, ...

Synopsis

This Operator generates a k-Nearest Neighbor model. It is used for classification or regression.

Set regression performance measures

Views: Design Results Turbo Prep Auto Model

Find data, operators...etc All Studio

Repository

- Import Data
- Community Samples (connected)
- Keras Samples
- DB (Legacy)
- Local Repository (admin)
 - Connections (admin)
 - 2019 (admin)
 - Business Data Mining (admin)
 - Lecture 5 Data
 - Lecture 6 Data

Process ▶ Cross Validation

1 Add <<Apply Model>>

2 Add <<Performance (Classification)>>

3 Check the followings:

- accuracy
- classification error

Parameters

% Performance (Performance (Classification))

main criterion: first

☒ accuracy

☒ classification error ✓

☐ kappa

☐ weighted mean recall

☐ weighted mean precision

☐ spearman rho

[Hide advanced parameters](#)

Help

% Performance (Classification)

RapidMiner Studio Core

Tags: [Accuracy](#), [Errors](#), [Precision](#), [Recall](#), [Kappa](#), [Squared](#), [Relative](#), [Validations](#), [Evaluations](#), [Metrics](#), [Predictive](#)

Synopsis

This operator is used for statistical

Run the analysis process and check the performance results

The screenshot displays the Orange3 data mining software interface. At the top, there is a toolbar with icons for file operations and a 'Views' section with buttons for 'Design', 'Results', 'Turbo Prep', and 'Auto Model'. A search bar on the right contains the text 'Find data, operators...etc' and a magnifying glass icon. Below the toolbar, a tab bar shows three open tabs: 'Result History', 'PerformanceVector (Performance)', and 'ExampleSet (Cross Validation)'. The 'KNNClassification (k-NN)' tab is currently selected, indicated by a light blue icon and a light blue background. The main workspace area is divided into two panes. The left pane, titled 'Description', contains the text: 'Weighted 5-Nearest Neighbour model for classification. The model contains 1599 examples with 11 dimensions of the following classes: 5, 6, 7, 4, 8, 3'. The right pane, titled 'Annotations', is empty. A callout box with a blue border and a blue circle containing the number '1' points to the 'KNNClassification (k-NN)' tab, with the text 'Click 'KNNClassification (k-NN)' to see the model'. Another callout box with a grey border and a grey background points to the 'Description' pane, with the text 'This shows that no special information.'

Views: Design Results Turbo Prep Auto Model

Find data, operators...etc

All Studio

Result History PerformanceVector (Performance) ExampleSet (Cross Validation) KNNClassification (k-NN)

KNNClassification

Description

Weighted 5-Nearest Neighbour model for classification.
The model contains 1599 examples with 11 dimensions of the following classes:
5
6
7
4
8
3

Annotations

This shows that no special information.

Click 'KNNClassification (k-NN)' to see the model

1

Run the analysis process and check the performance results

Result History

PerformanceVector (Performance)

ExampleSet (Cross Validation)

KNNClassification (k...

Views: Design Results Turbo Prep Auto Model

Click 'Example Set(Cross Validation)' to see prediction results.

1

examples): all

Open in Turbo Prep Auto Model

Row No.	quality	prediction(q...	confidence(8)	confidence(3)	confidence(6)	confidence(7)	confidence(4)	confidence(5)	fixed acidity	volatile acidity	citric acid	resic
1	5	5	0	0	0.372	0	0	0.628	7.900	0.600	0.060	1.60
2	5	6	0	0	0.593	0	0	0.407	5.600	0.615	0	1.60
3	5	5	0	0	0	0	0	1	8.900	0.620	0.180	3.80
4	6	5				0	0				0.510	1.80
5	6	7				0.587	0				0.140	2.40
6	5	5				0	0					1.90
7	5	5	0	0	0.388	0	0				0.070	2.40
8	6	5	0.205	0	0.180	0.208	0				0.140	2.40
9	5	5	0	0	0	0	0	1	7.300	0.450	0.360	5.90
10	5	6	0	0	0.587	0	0.194	0.219	7.500	0.630	0.120	5.10
11	6	5	0	0	0.206	0	0	0.794	7.300	0.390	0.310	2.40
12	5	6	0	0	0.804	0	0	0.196	7.700	0.690	0.220	1.90
13	5	5	0	0	0.221	0	0	0.779	9.700	0.320	0.540	2.50
14	6	6	0	0	0.589	0	0	0.411	6.300	0.300	0.480	1.80
15	5	5	0	0	0.218	0	0	0.782	7.700	0.490	0.260	1.90
16	5	5	0	0	0	0	0	1	7.900	0.520	0.260	1.90

This is a predicted label

This shows confidence level for the predicted label. Confidence for 6 is slightly huger than 5

ExampleSet (1,599 examples, 8 special attributes, 11 regular attributes)

Business Data Mining ©Kim Yang Sok

43

Run the analysis process and check the performance results

Views:

Design

Results

Turbo Prep

Auto Model

Find data, operators...etc

All Studio

Result History

PerformanceVector (Performance)

ExampleSet (Cross Validation)

KNNClassification (k-NN)

Criterion

accuracy

classification error

Performance

Description

Annotations

Table View

Plot View

accuracy: 53.35% +/- 2.31% (micro average: 53.35%)

	true 5	true 6	true 7	true 4	true 8	true 3	class precision
pred. 5	434	225	49	26	1	5	58.65%
pred. 6	206	352	78	20	11	3	52.54%
pred. 7	33	57	67	6	6	1	39.41%
pred. 4	7	2	3	0	0	1	0.00%
pred. 8	0	2	1	1	0	0	0.00%
pred. 3	1	0	1	0	0	0	0.00%
class recall	63.73%	55.17%	33.67%	0.00%	0.00%	0.00%	

Click 'Performance Vector(Performance)' to see performance results.

1

This confusion matrix shows detailed class precisions and recalls

Business Data Mining ©Kim Yang Sok

44

Exercise 4:

Choose the Best K and Similarity for K -NN

Exercise: Choose the Best K and Similarity for K-NN

Task & Process

- **Task**
 - Choose best parameters for K and Similarity automatically
- **Process**
 - Load “red wine” dataset
 - Split data into 70% training and 30% testing
 - Add Optimize Parameters(Grid)
 - Configure test design
 - Set parameters for the Optimize Parameters(Grid)
 - Run the analysis process and check the performance results

Load and split Data and add Optimize Parameters

Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators...etc All Studio

Repository

- Import Data
- Labor-Negotiations (v1)
- Market-Data (v1)
- Polynomial (v1)
- Products (v1)
- Purchases (v1)
- Ripley-Set (v1)
- Sonar (v1)
- Titanic (v1)**
- Titanic Training (v1)

Operators

Performance

- Performance (Classification)
- Performance (Binominal Cla
- Performance (Regression)**
- Performance (Costs)
- Performance (Ranking)
- Performance (Support Vecto

We found "Model Management" in the Marketplace. [Show me!](#)

Process

Process

100%

Process

Read CSV

Split Data

Optimize Parameter...

Parameters

Process

- logverbosity: init
- logfile: [Folder Icon]
- resultfile: [Folder Icon]
- random seed: 2001
- send mail: never
- encoding: SYSTEM
- [Hide advanced parameters](#)
- [Change compatibility \(9.5.001\)](#)

Help

Process

RapidMiner Studio Core




Synopsis



The root operator which is the outer most operator of every process.

Description



Recommended Operators


Design modelling process







Views: **Design** Results Turbo Prep Auto Model Deployments

Find data, operators...etc  All Studio 

Repository 





Labor-Negotiations (v1)

Market-Data (v1)

Polynomial (v1)

Products (v1)


Purchases (v1)


Ripley-Set (v1)


Sonar (v1)


Titanic (v1)


Titanic Training (v1)


Operators 


Performance 


 Performance (Classification)


 Performance (Binomial Classification)


 **Performance (Regression)**


 Performance (Costs)







 Performance (Ranking)

 Performance (Support Vector)

 We found "Model Management" in the Marketplace. [Show me!](#)

Process 

 **Process** ▶ Optimize Parameters (Grid) ▶

100%      


Optimize Parameters (Grid)


inp

inp


inp


k-NN

tra  mod
exa





Apply Model

mod  lab
unl mod



Performance


lab  per
per exa





per


mod

out


Recommended Operators 





Parameters 



 **Performance (Performance (Regression))**



main criterion


first 





☒ root mean squared error 


☐ absolute error  


☐ relative error  


☐ relative error lenient 

☐ relative error strict 

☐ normalized absolute error 

 [Hide advanced parameters](#)

Help 

 **Performance (Regression)**

RapidMiner Studio Core

Tags: [RMSE](#), [Errors](#), [Absolute](#), [Relative](#), [Squared](#), [Predictive](#)

Synopsis

This operator is used for statistical

Select parameters for optimization

Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators...etc All Studio

Repository

- Import Data
- Labor-Negotiations (v1)
- Market-Data (v1)
- Polynomial (v1)
- Products (v1)
- Purchases (v1)
- Ripley-Set (v1)
- Sonar (v1)
- Titanic (v1)**
- Titanic Training (v1)

Operators

- Performance (Classification)
- Performance (Binominal Cla)
- Performance (Regression)**
- Performance (Costs)
- Performance (Ranking)
- Performance (Support Vecto)

Process

Select Parameters: configure operator

Select Parameters: **configure operator**
Configure this operator by means of a Wizard.

Operators

- k-NN (k-NN)**
- Apply Model (Apply Model)
- Performance (Performance (Regression))

Parameters

- k**
- weighted_vote
- measure_types
- mixed_measure
- nominal_measure
- numerical_measure
- divergence
- kernel_type

Selected Parameters

Grid/Range

Min	Max	Steps	Scale
1.0	100.0	10	linear

Value List

1
11
21
31
41

0 parameter / 1 combinations selected

Grid List

OK Cancel

Recommended Operators

Parameters

Optimize Parameters (Grid)

Edit Parameter Settings...

error handling fail on error

☒ log performance

☐ log all criteria

☐ synchronize

☒ enable parallel execution

[Hide advanced parameters](#)

[Change compatibility \(9.5.001\)](#)

Help

Optimize Parameters (Grid)

Concurrency

Tags: Iterate, Settings, Grid, Search, Tune, Optimal, Parameters

Synopsis

This Operator finds the optimal values

Set K for optimization

Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators...etc All Studio

Repository

- Import Data
- Labor-Negotiations (v1)
- Market-Data (v1)
- Polynomial (v1)
- Products (v1)
- Purchases (v1)
- Ripley-Set (v1)
- Sonar (v1)
- Titanic (v1)
- Titanic Training (v1)

Process

Select Parameters: configure operator

Select Parameters: **configure operator**
Configure this operator by means of a Wizard.

Operators

- k-NN (k-NN)
- Apply Model (Apply Model)
- Performance (Performance (Regression))

Parameters

- weighted_vote
- measure_types
- mixed_measure
- nominal_me
- divergence
- kernel_type
- kernel_gamr
- kernel_sigma

Selected Parameters

- k-NN.k
- k-NN.numerical_measure

Grid/Range

Min	Max	Steps	Scale
1.0	21	10	linear

Value List

1
3
5
7
9

☒ Grid ☐ List 2 parameters / 143 combinations selected

☒ OK ☐ Cancel

Parameters

Optimize Parameters (Grid)

Edit Parameter Settings...

error handling fail on error

☒ log performance

☐ log all criteria

☐ synchronize

☒ enable parallel execution

Hide advanced parameters

Change compatibility (9.5.001)

Help

Optimize Parameters (Grid)

Concurrency

Tags: Iterate, Settings, Grid, Search, Tune, Optimal, Parameters

Synopsis

This Operator finds the optimal values

We found "Model Management" in the Marketplace. [Show me!](#)

Set numerical_measure for optimization

Select Parameters: configure operator

Select Parameters: **configure operator**
Configure this operator by means of a Wizard.

Operators

- k-NN (k-NN)
- Apply Model (Apply Model)
- Performance (Performance (Regression))

Parameters

Selected Parameters

- k-NN.k
- k-NN.numerical_measure

Grid/Range

Min	Max	Steps	Scale
0.0	0.0	0	linear

Value List

- EuclideanDistance
- CosineSimilarity
- ManhattanDistance
- OverlapSimilarity
- CamberraDistance
- ChebychevDistance
- CorrelationSimilarity
- DiceSimilarity
- DynamicTimeWarpingDistance

2 parameters / 44 combinations selected

Recommended Operators

Parameters

Optimize Parameters (Grid)

error handling fail on error

☒ log performance

☐ log all criteria

☐ synchronize

☒ enable parallel execution

[Hide advanced parameters](#)

[Change compatibility \(9.5.001\)](#)

Optimize Parameters (Grid)

Concurrency






Tags: [Iterate](#), [Settings](#), [Grid](#), [Search](#), [Tune](#), [Optimal](#), [Parameters](#)

Synopsis


This Operator finds the optimal values

We found "Model Management" in the Marketplace. [Show me!](#)

Execute the process and examine the results



Views: Design **Results** Turbo Prep Auto Model Deployments

 All Studio ▾

Result History

ParameterSet (Optimize Parameters (Grid))

PerformanceVector (Performance)

Optimize Parameters (Grid)

Description

Annotations

ParameterSet

Parameter set:

Performance:

PerformanceVector [

-----root_mean_squared_error: 0.697 +/- 0.000






]

k-NN.k = 7

k-NN.numerical_measure = EuclideanDistance

<new process*> - RapidMiner Studio Educational 9.5.001 @ DESKTOP-TPCESUJ

File Edit Process View Connections Settings Extensions Help



Views: Design **Results** Turbo Prep Auto Model Deployments

Result History

ParameterSet (Optimize Parameters (Grid))

PerformanceVector (Performance)

Optimize Parameters (Grid)

Data

Simple Charts

Advanced Charts

Optimize Parameters (Grid) (44 rows, 4 columns)

iteration	k-NN.k	k-NN.numerical_measure	root_mean_squared_error ↑
4	7	EuclideanDistance	0.697
15	7	CosineSimilarity	0.697
26	7	ManhattanDistance	0.697
37	7	OverlapSimilarity	0.697
5	9	EuclideanDistance	0.700
16	9	CosineSimilarity	0.700
27	9	ManhattanDistance	0.700
38	9	OverlapSimilarity	0.700
3	5	EuclideanDistance	0.701
14	5	CosineSimilarity	0.701
25	5	ManhattanDistance	0.701
36	5	OverlapSimilarity	0.701
8	15	EuclideanDistance	0.703

Exercise 5:

Explain Model Prediction

Exercise: Choose the Best K and Similarity for K-NN

Task & Process

- **Task**
 - Explain prediction results by using a model explainer
- **Process**
 - Load “red wine” dataset
 - Split data into 70% training and 30% testing
 - Add k-NN
 - Add Explain Prediction and connect with k-NN and Split Data
 - Run the analysis process and check the explaining results

Load dataset and spit data and add Explain Predictions

The screenshot displays the RapidMiner Studio interface with a workflow designed to load a dataset, split it, and explain predictions using a k-NN model. A blue callout box highlights the configuration for the k-NN operator: "Set k=7, measure types=NumricalMeasures and numerical measure=EuclideanDistance".

Repository: Lists datasets including Labor-Negotiations (v1), Market-Data (v1), Polynomial (v1), Products (v1), Purchases (v1), Ripley-Set (v1), Sonar (v1), Titanic (v1), and Titanic Training (v1).

Operators: The "Explain" search filter is active, showing the "Explain Predictions" operator under the "Scoring (1)" category.

Process Diagram:

- Read CSV:** Imports data from a file.
- Split Data:** Splits the data into training (tra) and testing (tes) sets.
- k-NN:** Trains a k-Nearest Neighbors model on the training set.
- Explain Predictions:** Generates explanations for the model's predictions on the testing set.




Parameters:




- Process:** logverbosity (init), logfile, resultfile, random seed (2001), send mail (never), encoding (SYSTEM).
- Explain Predictions:** mod (selected), tra, tes, vis, exa, imp, glo.

Help: Provides a synopsis of the "Process" operator: "The root operator which is the outer most operator of every process."

Recommended Operators: A section at the bottom for finding additional operators.

Examine explanation





Views:



Design


Results


Turbo Prep



Auto Model

Deployments



Find data, operators...etc  All Studio 


ExampleSet (Explain Predictions) 

Result #2 

 **ExplainPredictionsIOObject (Explain Predictions)** 

Result History




 AttributeWeights (Explain Predictions) 



 Explanations

Row No.	quality	predict...	fixed a...	volatile ...	citric ...	residu...	chlorid...	free s...	total s...	density	pH	sulpha...	alcohol
1	5	5.569	7.400	0.700	0	1.900	0.076	11	34	0.998	3.510	0.560	9.400
2	5	5.429	7.800	0.880	0	2.600	0.098	25	67	0.997	3.200	0.680	9.800
3	5	5.569	7.400	0.700	0	1.900	0.076	11	34	0.998	3.510	0.560	9.400
4	5	5.422	7.400	0.660	0	1.800	0.075	13	40	0.998	3.510	0.560	9.400
5	5	5.278	7.900	0.600	0.060	1.600	0.069	15	59	0.996	3.300	0.460	9.400
6	7	5.286	7.800	0.580	0.020	2	0.073	9	18	0.997	3.360	0.570	9.500
7	5	5.549	7.500	0.500	0.360	6.100	0.071	17	102	0.998	3.350	0.800	10.500
8	5	5	8.900	0.620	0.180	3.800	0.176	52	145	0.999	3.160	0.880	9.200
9	5	5	8.900	0.620	0.190	3.900	0.170	51	148	0.999	3.170	0.930	9.200
10	5	5.147	8.100	0.560	0.280	1.700	0.368	16	56	0.997	3.110	1.280	9.300
11	5	5	7.900	0.430	0.210	1.600	0.106	10	37	0.997	3.170	0.910	9.500
12	5	5.129	8.500	0.490	0.110	2.300	0.084	9					
13	6	5.413	6.900	0.400	0.140	2.400	0.085	21					
14	6	5.857	7.800	0.645	0	2	0.082	8					
15	5	5.134	8.300	0.655	0.120	2.300	0.083	15					
16	5	5.142	5.200	0.320	0.250	1.800	0.103	13					


The table highlights the attributes that most strongly support (green) or contradict (red) each prediction.

Examine explanation






Views: Design **Results** Turbo Prep Auto Model Deployments


 All Studio ▾


ExampleSet (Explain Predictions) ×


Result #2 ×


 ExplainPredictionsIOObject (Explain Predictions) ×


Result History



 AttributeWeights (Explain Predictions) ×

 Data

 Statistics

 Visualizations

 Annotations

Open in  Turbo Prep  Auto Model

Filter (5,280 / 5,280 examples): all ▾

Row No.	Row No	Name	Value	Importance
1	1	fixed acidity	7.4	0.044
2	1	volatile acidity	0.7	0.040
3	1	citric acid	0.0	0.030
4	1	residual sugar	1.9	0.008
5	1	chlorides	0.076	-0.018
6	1	free sulfur dio...	11.0	0.270
7	1	total sulfur di...	34.0	-0.216
8	1	density	0.9978	-0.005
9	1	pH	3.51	-0.043
10	1	sulphates	0.56	0.051
11	1	alcohol	9.4	0.038
12	2	fixed acidity	7.8	-0.018
13	2	volatile acidity	0.88	-0.077
14	2	citric acid	0.0	-0.015

ExampleSet (5,280 examples, 0 special attributes, 4 regular attributes)

The table explains the reason for the prediction.

ExampleSet (Explain Predictions)

Result #2

ExplainPredictionsIOObject (Explain Predictions)

Result History

AttributeWeights (Explain Predictions)



Data



Weight

Visualizations



Annotations

attribute	weight ↓
free sulfur dioxide	0.196
alcohol	0.093
fixed acidity	0.071
sulphates	0.027
citric acid	0.018
chlorides	0.017
residual sugar	0.014
pH	0.013
density	0.012
volatile acidity	0.011
total sulfur dioxide	0.009

The table explains the importance of attributes in the model prediction.



QUESTIONS?