

# ML\_Core/ Interfaces/ IClassify

---

[Go Up](#)

## IMPORTS

ML\_Core | ML\_Core.Types |

## DESCRIPTIONS

### **MODULE** IClassify

	IClassify
--	-----------

Interface definition for Classification. Actual implementation modules will probably take parameters.

### Children

1. [GetModel](#) : Calculate the model to fit the observation data to the observed classes
2. [Classify](#) : Classify the observations using a model
3. [Report](#) : Report the confusion matrix for the classifier and training data

---

### **FUNCTION** GetModel

[IClassify](#) \

<b>DATASET</b> (Types.Layout_Model)	<b>GetModel</b>
(DATASET(Types.NumericField) observations, DATASET(Types.DiscreteField) classifications)	

Calculate the model to fit the observation data to the observed classes.

**PARAMETER** observations the observed explanatory values

**PARAMETER** classifications the observed classification used to build the model

**RETURN** the encoded model

## FUNCTION Classify

IClassify \

<b>DATASET</b> (Types.Classify_Result)	<b>Classify</b>
(DATASET(Types.Layout_Model) model, DATASET(Types.NumericField) new_observations)	

Classify the observations using a model.

**PARAMETER** model The model, which must be produced by a corresponding getModel function.

**PARAMETER** new\_observations observations to be classified

**RETURN** Classification with a confidence value

## FUNCTION Report

IClassify \

<b>DATASET</b> (Types.Confusion_Detail)	<b>Report</b>
(DATASET(Types.Layout_Model) model, DATASET(Types.NumericField) observations, DATASET(Types.DiscreteField) classifications)	

Report the confusion matrix for the classifier and training data.

**PARAMETER** model the encoded model

**PARAMETER** observations the explanatory values.

**PARAMETER** classifications the classifications associated with the observations

**RETURN** the confusion matrix showing correct and incorrect results

---