

root

[Go Up](#)

Name	LogisticRegression
Version	3.0.0
Description	Logistic Regression implementation
License	http://www.apache.org/licenses/LICENSE-2.0
Copyright	Copyright (C) 2017 HPCC Systems
Authors	HPCCSystems
DependsOn	ML_Core 3.2.1, PBblas
Platform	6.2.0

Table of Contents

BinomialConfusion.ecl
Calculate the binomial confusion matrix
BinomialLogisticRegression.ecl
Binomial logistic regression using iteratively re-weighted least squares
Confusion.ecl
Generate the confusion matrix, to compare actual versus predicted response variable values
Constants.ecl
Constants used by Logistic Regression
DataStats.ecl
Produce summary information about the datasets
Deviance_Analysis.ecl
Analysis of Deviance Report
Deviance_Detail.ecl
Deviance detail report
dimm.ecl
Matrix multiply when either A or B is a diagonal and is passed as a vector
enum_workitems.ecl

Create an enumeration of string contents to be used as work items
ExtractBeta.ecl Extract the beta values form the model dataset
ExtractBeta_CI.ecl Extract the beta values and confidence intervals from the model dataset
ExtractBeta_full.ecl Extract the coefficient information including confidence intervals, z and p values
ExtractBeta_pval.ecl Extract the beta values including z and p value from the model
ExtractReport.ecl Create a model report from a model
LogitPredict.ecl Predict the category values with the logit function and the the supplied beta coefficients
LogitScore.ecl Calculate the score using the logit function and the the supplied beta coefficients
LUCI_Model.ecl Create a LUCI model file description of the model(s) from the external version of the model
Model_Deviance.ecl Model Deviance Report
Named_Model.ecl Apply external labels for work items and field names to a model
Null_Deviance.ecl Return Deviance information for the null model, that is, a model with only an intercept
Types.ecl Type definitions for LogisticRegression bundle

BinomialConfusion

[Go Up](#)

IMPORTS

ML_Core.Types | Types |

DESCRIPTIONS

FUNCTION BinomialConfusion

<code>DATASET(Types.Binomial_Confusion_Summary)</code>	BinomialConfusion
<code>(DATASET(Core_Types.Confusion_Detail) d)</code>	

Calculate the binomial confusion matrix. Work items with multinomial responses are ignored by this function. The higher value lexically is considered to be the positive indication.

PARAMETER `d` ||| TABLE (Confusion_Detail) — confusion detail for the work item and classifier.

RETURN TABLE ({ UNSIGNED2 wi , UNSIGNED4 classifier , UNSIGNED8 true_positive , UNSIGNED8 true_negative , UNSIGNED8 false_positive , UNSIGNED8 false_negative , UNSIGNED8 cond_pos , UNSIGNED8 pred_pos , UNSIGNED8 cond_neg , UNSIGNED8 pred_neg , REAL8 prevalence , REAL8 accuracy , REAL8 true_pos_rate , REAL8 false_neg_rate , REAL8 false_pos_rate , REAL8 true_neg_rate , REAL8 pos_pred_val , REAL8 false_disc_rate , REAL8 false_omit_rate , REAL8 neg_pred_val }) — confusion matrix for a binomial classifier in Binomial_Confusion_Summary format.

SEE Types.Binomial_Confusion_Summary

BinomialLogisticRegression

[Go Up](#)

IMPORTS

Constants | ML_Core.Interfaces | ML_Core.Types |

DESCRIPTIONS

MODULE BinomialLogisticRegression

BinomialLogisticRegression
(UNSIGNED max_iter=200, REAL8 epsilon=Constants.default_epsilon, REAL8 ridge=Constants.default_ridge)

Binomial logistic regression using iteratively re-weighted least squares.

PARAMETER max_iter ||| UNSIGNED8 — (Optional) The maximum number of iterations to try.
Default = 200.

PARAMETER epsilon ||| REAL8 — (Optional) The minimum change in the Beta value estimate to continue

PARAMETER ridge ||| REAL8 — (Optional) A value to populate a diagonal matrix that is added to a matrix help assure that the matrix is invertible.

PARENT ML_Core.Interfaces.IClassify
</home/tetrapod/pccsource/ML_Core/Interfaces/IClassify.ecl>

Children

1. `GetModel` : Calculate the model to fit the observation data to the observed classes
2. `Classify` : Classify the observations using a model as previously returned from `GetModel`
3. `Report` : Report the confusion matrix for the classifier and training data

FUNCTION `GetModel`

`BinomialLogisticRegression` \

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

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

PARAMETER `classifications` ||| `TABLE (DiscreteField)` — the observed classification used to build the model in `DiscreteField` format.

PARAMETER `observations` ||| `TABLE (NumericField)` — the observed explanatory values in `NumericField` format.

RETURN `TABLE ({ UNSIGNED2 wi , UNSIGNED8 id , UNSIGNED4 number , REAL8 value })` — the encoded model in `Layout_Model` format.

OVERRIDE

SEE `ML_Core.Types.NumericField`

SEE `ML_Core.Types.DiscreteField`

SEE `ML_Core.Types.Layout_Model`

FUNCTION `Classify`

`BinomialLogisticRegression` \

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

Classify the observations using a model as previously returned from GetModel.

PARAMETER `new_observations` ||| TABLE (NumericField) — observations to be classified in NumericField format.

PARAMETER `model` ||| TABLE (Layout_Model) — The model in Layout_Model format.

RETURN TABLE ({ UNSIGNED2 `wi` , UNSIGNED8 `id` , UNSIGNED4 `number` , INTEGER4 `value` , REAL8 `conf` }) — Classification with a confidence value in Classify_Result format.

OVERRIDE

SEE `ML_Core.Types.Layout_Model`

SEE `ML_Core.Types.NumericField`

SEE `ML_Core.Types.Classify_Result`

FUNCTION Report

[BinomialLogisticRegression](#) \

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

Report the confusion matrix for the classifier and training data.

PARAMETER `classifications` ||| TABLE (DiscreteField) — the actual classifications associated with the observations (i.e. ground truth) in DiscreteField format.

PARAMETER `model` ||| TABLE (Layout_Model) — the encoded model as returned from GetModel.

PARAMETER `observations` ||| TABLE (NumericField) — the explanatory values in NumericField format.

RETURN TABLE ({ UNSIGNED2 `wi` , UNSIGNED4 `classifier` , INTEGER4 `actual_class` , INTEGER4 `predict_class` , UNSIGNED4 `occurs` , BOOLEAN `correct` , REAL8 `pctActual` , REAL8 `pctPred` }) — the confusion matrix showing correct and incorrect results in Confusion_Detail format.

OVERRIDE

SEE `ML_Core.Types.NumericField`

SEE `ML_Core.Types.DiscreteField`

SEE `ML_Core.Types.ConfusionDetail`

Confusion

[Go Up](#)

IMPORTS

ML_Core | ML_Core.Types | Types |

DESCRIPTIONS

FUNCTION Confusion

<code>DATASET(Confusion_Detail)</code>	Confusion
<code>(DATASET(DiscreteField) dependents, DATASET(DiscreteField) predicts)</code>	

Generate the confusion matrix, to compare actual versus predicted response variable values.

PARAMETER `predicts` ||| TABLE (DiscreteField) — the predicted responses.

PARAMETER `dependents` ||| TABLE (DiscreteField) — the original response values.

RETURN TABLE ({ UNSIGNED2 `wi` , UNSIGNED4 `classifier` , INTEGER4 `actual_class` , INTEGER4 `predict_class` , UNSIGNED4 `occurs` , BOOLEAN `correct` , REAL8 `pctActual` , REAL8 `pctPred` }) — confusion matrix in Confusion_Detail format.

SEE ML_Core.Types.Confusion_Detail

Constants

[Go Up](#)

DESCRIPTIONS

MODULE Constants

Constants

Constants used by Logistic Regression. Most of these are the nominal values used by the Model data set. A few are used to control behavior.

Children

1. [limit_card](#) : No Documentation Found
2. [default_epsilon](#) : No Documentation Found
3. [default_ridge](#) : No Documentation Found
4. [local_cap](#) : No Documentation Found
5. [id_base](#) : No Documentation Found
6. [id_iters](#) : No Documentation Found
7. [id_delta](#) : No Documentation Found
8. [id_correct](#) : No Documentation Found
9. [id_incorrect](#) : No Documentation Found
10. [id_stat_set](#) : No Documentation Found
11. [id_betas](#) : No Documentation Found
12. [id_betas_coef](#) : No Documentation Found
13. [id_betas_SE](#) : No Documentation Found
14. [base_builder](#) : No Documentation Found

- 15. [base_max_iter](#) : No Documentation Found
 - 16. [base_epsilon](#) : No Documentation Found
 - 17. [base_ind_vars](#) : No Documentation Found
 - 18. [base_dep_vars](#) : No Documentation Found
 - 19. [base_obs](#) : No Documentation Found
 - 20. [builder_irls_local](#) : No Documentation Found
 - 21. [builder_irls_global](#) : No Documentation Found
 - 22. [builder_softmax](#) : No Documentation Found
-

ATTRIBUTE `limit_card`

[Constants](#) \

UNSIGNED2	<code>limit_card</code>
------------------	-------------------------

No Documentation Found

RETURN UNSIGNED2 —

ATTRIBUTE `default_epsilon`

[Constants](#) \

REAL8	<code>default_epsilon</code>
--------------	------------------------------

No Documentation Found

RETURN REAL8 —

ATTRIBUTE default_ridge

[Constants](#) \

REAL8	default_ridge
--------------	---------------

No Documentation Found

RETURN REAL8 —

ATTRIBUTE local_cap

[Constants](#) \

UNSIGNED4	local_cap
------------------	-----------

No Documentation Found

RETURN UNSIGNED4 —

ATTRIBUTE id_base

[Constants](#) \

	id_base
--	---------

No Documentation Found

RETURN INTEGER8 —

ATTRIBUTE id_iters

[Constants](#) \

id_iters

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE id_delta

[Constants](#) \

id_delta

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE id_correct

[Constants](#) \

id_correct

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE id_incorrect

[Constants](#) \

id_incorrect

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE id_stat_set

[Constants](#) \

id_stat_set

No Documentation Found

RETURN SET (INTEGERS) —

ATTRIBUTE id_betas

[Constants](#) \

id_betas

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE id_betas_coef

[Constants](#) \

id_betas_coef

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE id_betas_SE

[Constants](#) \

id_betas_SE

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE base_builder

[Constants](#) \

base_builder

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE base_max_iter

[Constants](#) \

base_max_iter

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE base_epsilon

[Constants](#) \

base_epsilon

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE base_ind_vars

[Constants](#) \

base_ind_vars

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE base_dep_vars

[Constants](#) \

base_dep_vars

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE base_obs

[Constants](#) \

base_obs

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE builder_irls_local

[Constants](#) \

builder_irls_local

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE builder_irls_global

[Constants](#) \

builder_irls_global

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE builder_softmax

[Constants](#) \

builder_softmax

No Documentation Found

RETURN INTEGERS —

DataStats

[Go Up](#)

IMPORTS

LogisticRegression.Types | LogisticRegression.Constants | ML_Core.Types |

DESCRIPTIONS

FUNCTION DataStats

<code>DATASET(Core.Types.NumericField)</code>	<code>DataStats</code>
<code>(DATASET(Core.Types.NumericField) indep, DATASET(Core.Types.DiscreteField) dep, BOOLEAN field_details=FALSE)</code>	

Produce summary information about the datasets.

When `field_details = FALSE`, indicates the range for the x and y (independent and dependent) columns.

When `field_details = TRUE`, the cardinality, minimum, and maximum values are returned. A zero cardinality is returned when the field cardinality exceeds the `Constants.limit_card` value.

Note that a column of all zero values cannot be distinguished from a missing column.

PARAMETER `indep` ||| TABLE (NumericField) — data set of independent variables.

PARAMETER `dep` ||| TABLE (DiscreteField) — data set of dependent variables.

PARAMETER `field_details` ||| BOOLEAN — Boolean directive to provide field level info.

RETURN TABLE ({ UNSIGNED2 `wi` , UNSIGNED4 `dependent_fields` , UNSIGNED4 `dependent_records` , UNSIGNED4 `independent_fields` , UNSIGNED4

`independent_records , UNSIGNED4 dependent_count , UNSIGNED4
independent_count , TABLE (Field_Desc) dependent_stats , TABLE (Field_Desc
) independent_stats })` — a data set of information on each work item in Data_Info format.

SEE Types.Data_Info

SEE Constants.limit_card

Deviance_Analysis

[Go Up](#)

IMPORTS

Types | ML_Core.Math |

DESCRIPTIONS

FUNCTION Deviance_Analysis

<code>DATASET(Types.AOD_Record)</code>	Deviance_Analysis
<code>(DATASET(Types.Deviance_Record) proposed,</code> <code>DATASET(Types.Deviance_Record) base)</code>	

Analysis of Deviance Report.

Compare deviance information between two models, a base and and proposed model.

Analysis of Deviance is analogous to the Analysis of Variance (ANOVA) used in least-squares modeling, but adapted to the general linear model (GLM). In this case it is adapted specifically to the logistic model.

The inputs are the deviance records for each model as obtained from a call to Model_Deviance.

PARAMETER base ||| TABLE (Deviance_Record) — deviance records of the base model for comparison.

PARAMETER proposed ||| TABLE (Deviance_Record) — deviance records of the proposed model.

RETURN TABLE ({ UNSIGNED2 wi , UNSIGNED4 classifier , UNSIGNED8 residual_df , UNSIGNED8 df , REAL8 residual_dev , REAL8 deviance , REAL8 p_value }) — the comparison of the deviance between the models in AOD_Record format.

SEE Model_Deviance

SEE Types.Deviance_Record

SEE Types.AOD_Record

Deviance_Detail

[Go Up](#)

IMPORTS

ML_Core | ML_Core.Types | Types |

DESCRIPTIONS

FUNCTION Deviance_Detail

<code>DATASET(Types.Observation_Deviance)</code>	<code>Deviance_Detail</code>
<code>(DATASET(Core_Types.DiscreteField) dependents, DATASET(Types.Raw_Prediction) predicts)</code>	

Deviance detail report.

Provides deviance information for each observation.

Analysis of Deviance is analogous to the Analysis of Variance (ANOVA) used in least-squares modeling, but adapted to the general linear model (GLM). In this case it is adapted specifically to the logistic model.

PARAMETER `predicts` ||| TABLE (Raw_Prediction) — the predicted values of the response variable

PARAMETER `dependents` ||| TABLE (DiscreteField) — original dependent records for the model

RETURN TABLE ({ UNSIGNED2 wi , UNSIGNED8 id , UNSIGNED4 classifier , INTEGER4 actual , INTEGER4 predicted , REAL8 mod_ll , REAL8 mod_dev_component , REAL8 mod_dev_residual , REAL8 nil_ll , REAL8 nil_dev_component , REAL8 nil_dev_residual }) — the deviance information by observation and the log likelihood of the predicted result in Observation_Deviance format.

SEE Types.Observation_Deviance

dimmm

[Go Up](#)

IMPORTS

std.blas | std.BLAS.Types |

DESCRIPTIONS

EMBED dimmm

<code>Types.matrix_t</code>	dimmm
<code>(BOOLEAN transposeA, BOOLEAN transposeB, BOOLEAN diagonalA, BOOLEAN diagonalB, Types.dimension_t m, Types.dimension_t n, Types.dimension_t k, Types.value_t alpha, Types.matrix_t A, Types.matrix_t B, Types.value_t beta=0.0, Types.matrix_t C=[])</code>	

Matrix multiply when either A or B is a diagonal and is passed as a vector.

Computes: $\alpha * \text{op}(A) \text{op}(B) + \beta * C$ where $\text{op}()$ is transpose.

PARAMETER A ||| SET (REAL8) — matrix A.

PARAMETER k ||| UNSIGNED4 — number of columns/rows for the multiplier/multiplicand.

PARAMETER alpha ||| REAL8 — scalar used on A.

PARAMETER n ||| UNSIGNED4 — number of columns in product.

PARAMETER transposeA ||| BOOLEAN — true when transpose of A is used.

PARAMETER beta ||| REAL8 — scalar for matrix C.

PARAMETER C ||| SET (REAL8) — matrix C or empty.

PARAMETER m ||| UNSIGNED4 — number of rows in product.

PARAMETER diagonalB ||| BOOLEAN — true when B is the diagonal matrix.

PARAMETER B ||| SET (REAL8) — matrix B.

PARAMETER diagonalA ||| BOOLEAN — true when A is the diagonal matrix.

PARAMETER transposeB ||| BOOLEAN — true when transpose of B is used.

RETURN SET (REAL8) — result matrix in matrix_t format.

SEE Std.BLAS.Types.matrix_t

enum_workitems

[Go Up](#)

DESCRIPTIONS

MACRO `enum_workitems`

<code>enum_workitems</code>
<code>(dsIn, dsOut, src_field, wi_name)</code>

Create an enumeration of string contents to be used as work items.

This macro produces 2 external symbols, `dsOut` and `dsOut_Map`.

The `dsOut` extends the input dataset with a numeric work-item number.

The `dsOut_Map` dataset captures the relationship between the strings that name the work items and the nominal assigned in `Workitem_Mapping` format.

PARAMETER `dsOut` ||| INTEGER8 — the symbol to use for the appended data.

PARAMETER `wi_name` ||| INTEGER8 — the field name for the work item value assigned.

PARAMETER `src_field` ||| INTEGER8 — a field name to use to discriminate work-items.

PARAMETER `dsIn` ||| INTEGER8 — the input recordset.

RETURN — Nothing. The macro creates the symbols '`dsOut`' and '`dsOut_Map`' inline.

SEE `Types.Workitem_Mapping`

ExtractBeta

[Go Up](#)

IMPORTS

ML_Core.Types | Types |

DESCRIPTIONS

FUNCTION ExtractBeta

ExtractBeta
(DATASET(Core_Types.Layout_Model) mod_ds)

Extract the beta values form the model dataset.

PARAMETER mod_ds ||| TABLE (Layout_Model) — the model as returned from GetModel.

RETURN TABLE ({ UNSIGNED2 wi , UNSIGNED4 ind_col , UNSIGNED4 dep_nom , REAL8 w , REAL8 SE }) — the beta values as Model_Coef records, with zero as the constant term.

SEE Types.Model_Coef

ExtractBeta_CI

[Go Up](#)

IMPORTS

ML_Core.Types | Types |

DESCRIPTIONS

FUNCTION ExtractBeta_CI

<code>DATASET(Types.Confidence_Model_Coef)</code>	<code>ExtractBeta_CI</code>
<code>(DATASET(Core_Types.Layout_Model) mod_ds, REAL8 level)</code>	

Extract the beta values and confidence intervals from the model dataset.

PARAMETER `mod_ds` ||| TABLE (Layout_Model) — the model as returned from GetModel.

PARAMETER `level` ||| REAL8 — the significance value for the intervals.

RETURN TABLE ({ UNSIGNED2 `wi` , UNSIGNED4 `ind_col` , UNSIGNED4 `dep_nom` , REAL8 `w` , REAL8 `SE` , REAL8 `upper` , REAL8 `lower` }) — the beta values with confidence intervals in Confidence_Model_Coef format, with zero as the constant term.

SEE Types.Confidence_Model_Coef

ExtractBeta_full

[Go Up](#)

IMPORTS

ML_Core.Types | Types | ML_Core.Math |

DESCRIPTIONS

FUNCTION ExtractBeta_full

<code>DATASET(Types.Full_Model_Coef)</code>	<code>ExtractBeta_full</code>
<code>(DATASET(Core_Types.Layout_Model) mod_ds, REAL8 level=0.05)</code>	

Extract the coefficient information including confidence intervals, z and p values.

PARAMETER `mod_ds` ||| TABLE (Layout_Model) — the model as returned from GetModel.

PARAMETER `level` ||| REAL8 — the significance value for the intervals.

RETURN TABLE ({ UNSIGNED2 `wi` , UNSIGNED4 `ind_col` , UNSIGNED4 `dep_nom` , REAL8 `w` , REAL8 `SE` , REAL8 `z` , REAL8 `p_value` , REAL8 `upper` , REAL8 `lower` }) — the coefficient information for the model in Full_Model_Coef format, with zero as the constant term.

SEE Types.Full_Model_Coef

ExtractBeta_pval

[Go Up](#)

IMPORTS

ML_Core.Types | Types |

DESCRIPTIONS

FUNCTION ExtractBeta_pval

<code>DATASET(Types.pval_Model_Coef)</code>	<code>ExtractBeta_pval</code>
<code>(DATASET(Core_Types.Layout_Model) mod_ds)</code>	

Extract the beta values including z and p value from the model.

PARAMETER `mod_ds` ||| TABLE (Layout_Model) — the model as returned from GetModel.

RETURN TABLE ({ UNSIGNED2 wi , UNSIGNED4 ind_col , UNSIGNED4 dep_nom , REAL8 w , REAL8 SE , REAL8 z , REAL8 p_value }) — the beta values with p-values in pval_Model_Coef format, with zero as the constant term.

SEE Types.pval_Model_Coef

ExtractReport

[Go Up](#)

IMPORTS

ML_Core.Types | Types | Constants |

DESCRIPTIONS

FUNCTION ExtractReport

<code>DATASET(Types.Model_Report)</code>	ExtractReport
<code>(DATASET(Core_Types.Layout_Model) mod_ds)</code>	

Create a model report from a model.

PARAMETER `mod_ds` ||| TABLE (Layout_Model) — the model as returned from GetModel.

RETURN TABLE ({ UNSIGNED2 wi , UNSIGNED4 max_iterations , REAL8 epsilon , UNSIGNED4 dep_vars , UNSIGNED4 ind_vars , UNSIGNED8 obs , UNSIGNED2 builder , TABLE (Classifier_Stats) stats }) — the model report in Model_Report format.

SEE Types.Model_Report

LogitPredict

[Go Up](#)

IMPORTS

ML_Core.Types | Types |

DESCRIPTIONS

FUNCTION LogitPredict

<code>DATASET(Classify_Result)</code>	LogitPredict
<code>(DATASET(Model_Coef) coef, DATASET(NumericField) independents)</code>	

Predict the category values with the logit function and the the supplied beta coefficients.

PARAMETER `independents` ||| TABLE (NumericField) — the observations.

PARAMETER `coef` ||| TABLE (Model_Coef) — the model beta coefficients as returned from `ExtractBeta`.

RETURN TABLE ({ UNSIGNED2 `wi` , UNSIGNED8 `id` , UNSIGNED4 `number` , INTEGER4 `value` , REAL8 `conf` }) — the predicted category values and a confidence score in `Classify_Result` format.

SEE `ExtractBeta`

SEE `ML_Core.Types.Classify_Result`

LogitScore

[Go Up](#)

IMPORTS

`ML_Core.Types` | `Types` |

DESCRIPTIONS

FUNCTION `LogitScore`

<code>DATASET(Raw_Prediction)</code>	LogitScore
<code>(DATASET(Model_Coef) coef, DATASET(NumericField) independents)</code>	

Calculate the score using the logit function and the the supplied beta coefficients.

PARAMETER `independents` ||| `TABLE (NumericField)` — the observations.

PARAMETER `coef` ||| `TABLE (Model_Coef)` — the model beta coefficients as returned from `ExtractBetas`.

RETURN `TABLE ({ UNSIGNED2 wi , UNSIGNED8 id , UNSIGNED4 number , REAL8 raw })` — the raw prediction value in `Raw_Prediction` format.

SEE `ExtractBetas`

SEE `Types.Raw_Prediction`

LUCI_Model

[Go Up](#)

IMPORTS

Types | std.Str | std.system.ThorLib |

DESCRIPTIONS

FUNCTION LUCI_Model

<code>DATASET(Types.LUCI_Rec)</code>	LUCI_Model
<code>(DATASET(Types.LUCI_Model_Rqst) rqst, DATASET(Types.External_Model) mod, STRING wi_field='work_item')</code>	

Create a LUCI model file description of the model(s) from the external version of the model.

LUCI is a proprietary format used within LexisNexis.

The multi-score card per model case assumes that the score card selection is based solely upon the work item field. If this is not the case, the L1SE records will need to be patched.

The model id and name may have a "\$" character that is updated to match the work item when there are multiple models applied. If the strings do not have a "\$" character, the work item string is appended.

The score card name may have a "\$" character which is updated to match the work item. If the name is blank, the score card is named for the work item.

LUCI data fields may not contain comma characters. This function requires that the work item identification strings do not contain characters that need special handling for CSV data.

PARAMETER wi_field ||| STRING — the field name holding the work item identification string.

PARAMETER `rqst` ||| TABLE (LUCI_Model_Rqst) — the information to map work items to models in LUCI_Model_Rqst format.

PARAMETER `mod` ||| TABLE (External_Model) — the model with the external field names applied in External_Model format as returned from Named_Model.

RETURN TABLE ({ STRING line }) — The lines of the LUCI file in LUCI_Rec format.

SEE Types.External_Model

SEE Named_Model

SEE Types.LUCI_Model_Rqst

SEE Types.LUCI_Rec

Model_Deviance

[Go Up](#)

IMPORTS

Types |

DESCRIPTIONS

FUNCTION Model_Deviance

<code>DATASET(Types.Deviance_Record)</code>	<code>Model_Deviance</code>
<code>(DATASET(Types.Observation_Deviance) od, DATASET(Types.Model_Coef) mod)</code>	

Model Deviance Report.

Create a report of deviance information for a model.

Analysis of Deviance is analogous to the Analysis of Variance (ANOVA) used in least-squares modeling, but adapted to the general linear model (GLM). In this case it is adapted specifically to the logistic model.

PARAMETER `od` ||| TABLE (Observation_Deviance) — observation-deviance records, as obtained from a call to Deviance_Detail.

PARAMETER `mod` ||| TABLE (Model_Coef) — model co-efficients records, as obtained from a call to ExtractBeta.

RETURN TABLE ({ UNSIGNED2 wi , UNSIGNED4 classifier , UNSIGNED8 df , REAL8 deviance , REAL8 AIC }) — model deviance in Deviance_Record format.

SEE Deviance_Detail

SEE ExtractBeta

SEE Types.Deviance_Record

Named_Model

[Go Up](#)

IMPORTS

Types |

DESCRIPTIONS

FUNCTION Named_Model

<code>DATASET(Types.External_Model)</code>	<code>Named_Model</code>
<pre>(DATASET(Types.Layout_Model) mod_ds, DATASET(Types.FieldName_Mapping) expl_map, DATASET(Types.FieldName_Mapping) resp_map, DATASET(Types.WorkItem_mapping) wi_map=empty, REAL8 level=0.05)</pre>	

Apply external labels for work items and field names to a model.

Returns an expanded model that includes:

- coefficients
- z and p-values
- independent variable field names
- dependent variable field names
- work-item names

PARAMETER `wi_map` ||| TABLE (WorkItem_Mapping) — (optional) mapping of workitem strings to workitem nominals in FieldName_Mapping format.

PARAMETER `expl_map` ||| TABLE (`FieldName_Mapping`) — the relation of the explanatory or independent variables to the field names for those variables in `FieldName_Mapping` format.

PARAMETER `mod_ds` ||| TABLE (`Layout_Model`) — the model as returned from `GetModel`.

PARAMETER `level` ||| REAL8 — (optional) value for confidence intervals. Default = 0.05.

PARAMETER `resp_map` ||| TABLE (`FieldName_Mapping`) — the relation of the response variable column numbers to the field names in `FieldName_Mapping` format.

RETURN TABLE ({ `STRING` `work_item` , `STRING` `response_field` , `UNSIGNED2` `wi` , `UNSIGNED4` `dep_nom` , TABLE (`External_Coef`) `coef` }) — an expanded model in `External_Model` format.

SEE `Types.FieldName_Mapping`

SEE `Types.External_Model`

Null_Deviance

[Go Up](#)

IMPORTS

Types |

DESCRIPTIONS

FUNCTION Null_Deviance

<code>DATASET(Types.Deviance_Record)</code>	<code>Null_Deviance</code>
<code>(DATASET(Types.Observation_Deviance) od)</code>	

Return Deviance information for the null model, that is, a model with only an intercept.

Analysis of Deviance is analogous to the Analysis of Variance (ANOVA) used in least-squares modeling, but adapted to the general linear model (GLM). In this case it is adapted specifically to the logistic model.

PARAMETER `od` ||| TABLE (Observation_Deviance) — Observation Deviance record set as returned from Deviance_Detail.

RETURN TABLE ({ UNSIGNED2 `wi` , UNSIGNED4 `classifier` , UNSIGNED8 `df` , REAL8 `deviance` , REAL8 `AIC` }) — a data set of the null model deviances for each work item and classifier in Deviance_Record format.

SEE Types.Observation_Deviance

SEE Types.Deviance_Record

SEE Deviance_Detail



Types

[Go Up](#)

IMPORTS

ML_Core.Types |

DESCRIPTIONS

MODULE Types

Types

Type definitions for LogisticRegression bundle

Children

1. [AnyField](#) : No Documentation Found
2. [NumericField](#) : The NumericField layout defines a matrix of Real valued data-points
3. [DiscreteField](#) : The Discrete Field layout defines a matrix of Integer valued data-points
4. [Layout_Model](#) : No Documentation Found
5. [t_work_item](#) : No Documentation Found
6. [t_RecordID](#) : No Documentation Found
7. [t_FieldNumber](#) : No Documentation Found
8. [t_FieldReal](#) : No Documentation Found
9. [t_Discrete](#) : No Documentation Found
10. [t_Universe](#) : No Documentation Found

11. [Field_Desc](#) : Describe information about each field in a training set
 12. [Data_Info](#) : Describes information about a training dataset composed of independent and dependent columns
 13. [NumericField_U](#) : Record structure to add a 'Universe Number' to a NumericField allowing multiple independent NumericField matrixes within a work-item
 14. [DiscreteField_U](#) : Record structure to add a 'Universe Number' to a DiscreteField allowing multiple independent DiscreteField matrixes within a work-item
 15. [Layout_Column_Map](#) : Layout for a column map record that is used to remap column numbers
 16. [Classifier_Stats](#) : Statistics about the effectiveness of each classifier in a model
 17. [Model_Report](#) : Statistical information about a model
 18. [Binomial_Confusion_Summary](#) : Accuracy stats for binomial classifications
 19. [Model_Coef](#) : Model Coefficients
 20. [Confidence_Model_Coef](#) : Model Coefficients with confidence intervals
 21. [pval_Model_Coef](#) : Model coefficients with z and p-value
 22. [Full_Model_Coef](#) : Model coefficients with confidence intervals and p-value
 23. [External_Coef](#) : Model coefficients, confidence intervals, and p-value, plus independent field names, for each coefficient
 24. [External_Model](#) : Expanded version of a model with statistics and field names
 25. [Raw_Prediction](#) : Record for raw prediction without confidence information
 26. [Observation_Deviance](#) : Record to contain deviance information about each observation
 27. [Deviance_Record](#) : Record to hold deviance summary information about a model
 28. [AOD_Record](#) : Record to hold Analysis of Deviance (AOD) information for a model
 29. [FieldName_Mapping](#) : Layout used to hold the mapping between a field's number and its name
 30. [WorkItem_Mapping](#) : Layout used to hold the mapping between a work-item number and a textual name for that work-item
 31. [LUCI_Rec](#) : Layout to store the lines of a generated LUCI model file
 32. [LUCI_Model_Rqst](#) : Format for information to guide the generation of a LUCI file
-

RECORD AnyField

Types \

AnyField

No Documentation Found

FIELD number ||| UNSIGNED4 — No Doc

FIELD id ||| UNSIGNED8 — No Doc

FIELD wi ||| UNSIGNED2 — No Doc

RECORD NumericField

Types \

NumericField

The NumericField layout defines a matrix of Real valued data-points. It acts as the primary Dataset layout for interacting with most ML Functions. Each record represents a single cell in a matrix. It is most often used to represent a set of data-samples or observations, with the 'id' field representing the data-sample or observation, and the 'number' field representing the various fields within the observation.

FIELD value ||| REAL8 — The value of this cell in the matrix.

FIELD number ||| UNSIGNED4 — This field represents the matrix column number for this cell. It is also considered the field number of the observation

FIELD id ||| UNSIGNED8 — This field represents the row-number of this cell of the matrix. It is also considered the record-id for observations / data-samples.

FIELD wi ||| UNSIGNED2 — The work-item id, supporting the Myriad style interface. This allows multiple independent matrixes to be contained within a single dataset, supporting independent ML activities to be processed in parallel.

RECORD DiscreteField

Types \

DiscreteField

The Discrete Field layout defines a matrix of Integer valued data-points. It is similar to the NumericField layout above, except for only containing discrete (integer) values. It is typically used to convey the class-labels for classification algorithms.

FIELD value ||| INTEGER4 — The value of this cell in the matrix.

FIELD number ||| UNSIGNED4 — This field represents the matrix column number for this cell. It is also considered the field number of the observation

FIELD id ||| UNSIGNED8 — This field represents the row-number of this cell of the matrix. It is also considered the record-id for observations / data-samples.

FIELD wi ||| UNSIGNED2 — The work-item id, supporting the Myriad style interface. This allows multiple independent matrixes to be contained within a single dataset, supporting independent ML activities to be processed in parallel.

RECORD Layout_Model

Types \

Layout_Model

No Documentation Found

FIELD value ||| REAL8 — No Doc

FIELD number ||| UNSIGNED4 — No Doc

FIELD id ||| UNSIGNED8 — No Doc

FIELD wi ||| UNSIGNED2 — No Doc

ATTRIBUTE t_work_item

[Types \](#)

t_work_item

No Documentation Found

RETURN UNSIGNED2 —

ATTRIBUTE t_RecordID

[Types \](#)

t_RecordID

No Documentation Found

RETURN UNSIGNED8 —

ATTRIBUTE t_FieldNumber

[Types \](#)

t_FieldNumber

No Documentation Found

RETURN UNSIGNED4 —

ATTRIBUTE t_FieldReal

Types \

t_FieldReal

No Documentation Found

RETURN REAL8 —

ATTRIBUTE t_Discrete

Types \

t_Discrete

No Documentation Found

RETURN INTEGER4 —

ATTRIBUTE t_Universe

Types \

t_Universe

No Documentation Found

RETURN UNSIGNED1 —

RECORD Field_Desc

Types \

Field_Desc

Describe information about each field in a training set.

FIELD min_value ||| REAL8 — the minimum value for the field.

FIELD number ||| UNSIGNED4 — the column (feature) number.

FIELD cardinality ||| UNSIGNED4 — the number of unique values in the field.

FIELD max_value ||| REAL8 — the maximum value for the field.

RECORD Data_Info

Types \

Data_Info

Describes information about a training dataset composed of independent and dependent columns.

FIELD dependent_records ||| UNSIGNED4 — the number of records in the dependent data.

FIELD wi ||| UNSIGNED2 — the work-item number.

FIELD independent_fields ||| UNSIGNED4 — the number of fields in the independent data.

FIELD dependent_stats ||| TABLE (Field_Desc) — dataset of Field_Desc records describing each of the fields of the dependent data.

FIELD independent_records ||| UNSIGNED4 — the number of records in the independent data.

FIELD independent_stats ||| TABLE (Field_Desc) — dataset of Field_Desc records describing each of the fields of the independent data.

FIELD dependent_fields ||| UNSIGNED4 — the number of fields in the dependent data.

FIELD dependent_count ||| UNSIGNED4 — No Doc

FIELD independent_count ||| UNSIGNED4 — No Doc

SEE Field_Desc

RECORD NumericField_U

Types \

NumericField_U

Record structure to add a 'Universe Number' to a NumericField allowing multiple independent NumericField matrixes within a work-item.

FIELD u ||| UNSIGNED1 — the 'universe' number identifying a distinct matrix within a NumericField dataset and work-item.

FIELD value ||| REAL8 — No Doc

FIELD number ||| UNSIGNED4 — No Doc

FIELD id ||| UNSIGNED8 — No Doc

FIELD wi ||| UNSIGNED2 — No Doc

RECORD DiscreteField_U

Types \

DiscreteField_U

Record structure to add a 'Universe Number' to a DiscreteField allowing multiple independent DiscreteField matrixes within a work-item.

FIELD u ||| UNSIGNED1 — the 'universe' number identifying a distinct matrix within a DiscreteField dataset and work-item.

FIELD value ||| INTEGER4 — No Doc

FIELD number ||| UNSIGNED4 — No Doc

FIELD id ||| UNSIGNED8 — No Doc

FIELD wi ||| UNSIGNED2 — No Doc

RECORD Layout_Column_Map

Types \

Layout_Column_Map

Layout for a column map record that is used to remap column numbers.

FIELD remap_number ||| UNSIGNED4 — the mapped-to field number.

FIELD orig_number ||| UNSIGNED4 — the original field number.

FIELD wi ||| UNSIGNED2 — the work-item number.

RECORD Classifier_Stats

Types \

Classifier_Stats

Statistics about the effectiveness of each classifier in a model.

FIELD incorrect ||| UNSIGNED4 — the number of classes predicted incorrectly in the training data.

FIELD correct ||| UNSIGNED4 — the number of classes predicted correctly in the training data.

FIELD max_delta ||| REAL8 — the max_delta value for the classifier.

FIELD column ||| UNSIGNED4 — the classifier field number.

FIELD iterations ||| UNSIGNED4 — the number of iterations used to train the classifier.

RECORD Model_Report

Types \

Model_Report

Statistical information about a model.

One record is generated per work-item.

FIELD ind_vars ||| UNSIGNED4 — the number of independent variables (i.e. features).

FIELD stats ||| TABLE (Classifier_Stats) — child dataset of Classifier_Stats, one for each classifier in the work-item.

FIELD dep_vars ||| UNSIGNED4 — the number of dependent variables (i.e. classifiers).

FIELD builder ||| UNSIGNED2 — the identifier for the builder used to train the model.

FIELD max_iterations ||| UNSIGNED4 — the maximum iterations use to train the model.

FIELD obs ||| UNSIGNED8 — the number of observations (i.e. records) in the training data.

FIELD epsilon ||| REAL8 — the 'epsilon' value used within the model.

FIELD wi ||| UNSIGNED2 — the work-item

SEE Classifier_Stats

RECORD Binomial_Confusion_Summary

Types \

Binomial_Confusion_Summary

Accuracy stats for binomial classifications.

One record per work-item and classifier.

FIELD classifier ||| UNSIGNED4 — the classifier field number (i.e. dependent field number).

- FIELD** true_positive ||| UNSIGNED8 — the count of true positive results (i.e. predicted = TRUE, actual = TRUE).
- FIELD** cond_neg ||| UNSIGNED8 — the count of results where actual = FALSE.
- FIELD** false_disc_rate ||| REAL8 — false_positive / pred_pos.
- FIELD** true_neg_rate ||| REAL8 — true_negative / cond_neg.
- FIELD** accuracy ||| REAL8 — (true_positive + true_negative) / total.
- FIELD** false_negative ||| UNSIGNED8 — the count of false_negative results (i.e. predicted = FALSE, actual = TRUE).
- FIELD** true_pos_rate ||| REAL8 — true_positive / cond_pos.
- FIELD** true_negative ||| UNSIGNED8 — the count of true negative results (i.e. predicted = FALSE, actual = FALSE).
- FIELD** neg_pred_val ||| REAL8 — true_negative / pred_neg.
- FIELD** pred_neg ||| UNSIGNED8 — the count of results where predicted = FALSE.
- FIELD** false_pos_rate ||| REAL8 — false_positive / cond_neg.
- FIELD** cond_pos ||| UNSIGNED8 — the count of results where actual = TRUE.
- FIELD** prevalence ||| REAL8 — cond_pos / total.
- FIELD** wi ||| UNSIGNED2 — the work-item number.
- FIELD** false_omit_rate ||| REAL8 — false_negative / pred_neg.
- FIELD** false_positive ||| UNSIGNED8 — the count of false_positive results (i.e. predicted = TRUE, actual = FALSE).
- FIELD** pred_pos ||| UNSIGNED8 — the count of results where predicted = TRUE.
- FIELD** pos_pred_val ||| REAL8 — true_positive / pred_pos.
- FIELD** false_neg_rate ||| REAL8 — No Doc

RECORD Model_Coef

Types \

Model_Coef

Model Coefficients.

FIELD dep_nom ||| UNSIGNED4 — the dependent column number (i.e. classifier number).

FIELD ind_col ||| UNSIGNED4 — the independent column number (i.e. feature number).

FIELD SE ||| REAL8 — the Standard Error of the coefficient.

FIELD w ||| REAL8 — the learned weight (i.e. coefficient).

FIELD wi ||| UNSIGNED2 — the work-item number.

RECORD Confidence_Model_Coef

Types \

Confidence_Model_Coef

Model Coefficients with confidence intervals.

FIELD lower ||| REAL8 — the lower range of the confidence interval

FIELD upper ||| REAL8 — the upper range of the confidence interval

FIELD se ||| REAL8 — No Doc

FIELD wi ||| UNSIGNED2 — No Doc

FIELD dep_nom ||| UNSIGNED4 — No Doc

FIELD w ||| REAL8 — No Doc

FIELD ind_col ||| UNSIGNED4 — No Doc

RECORD pval_Model_Coef

Types \

pval_Model_Coef

Model coefficients with z and p-value.

FIELD p_value ||| REAL8 — the p_value of the coefficient.

FIELD z ||| REAL8 — the z value.

FIELD se ||| REAL8 — No Doc

FIELD wi ||| UNSIGNED2 — No Doc

FIELD dep_nom ||| UNSIGNED4 — No Doc

FIELD w ||| REAL8 — No Doc

FIELD ind_col ||| UNSIGNED4 — No Doc

RECORD Full_Model_Coef

Types \

Full_Model_Coef

Model coefficients with confidence intervals and p-value

FIELD p_value ||| REAL8 — the p_value of the coefficient.

FIELD lower ||| REAL8 — the lower range of the confidence interval

FIELD upper ||| REAL8 — the upper range of the confidence interval

FIELD z ||| REAL8 — the z value.

FIELD se ||| REAL8 — No Doc

FIELD wi ||| UNSIGNED2 — No Doc

FIELD dep_nom ||| UNSIGNED4 — No Doc

FIELD w ||| REAL8 — No Doc

FIELD ind_col ||| UNSIGNED4 — No Doc

RECORD External_Coef

Types \

External_Coef

Model coefficients, confidence intervals, and p-value, plus independent field names, for each coefficient.

FIELD lower ||| REAL8 — the lower bound of the confidence interval.

FIELD ind_col ||| UNSIGNED4 — the field number of the independent field for this coefficient.

FIELD upper ||| REAL8 — the upper bound of the confidence interval.

FIELD z ||| REAL8 — the z value.

FIELD p_value ||| REAL8 — the p-value.

FIELD field_name ||| STRING — the name of the independent field for this coefficient.

FIELD isIntercept ||| BOOLEAN — Boolean field is TRUE if this is the intercept coefficient, otherwise FALSE.

FIELD w ||| REAL8 — the coefficient value (weight)

FIELD SE ||| REAL8 — the Standard Error of the coefficient

RECORD External_Model

Types \

External_Model

Expanded version of a model with statistics and field names.

Field names include independent data field names, dependent data field names and work-item names.

FIELD dep_nom ||| UNSIGNED4 — the field number of the classifier (i.e. dependent field number).

FIELD work_item ||| STRING — the work-item's name.

FIELD wi ||| UNSIGNED2 — the work-item number.

FIELD coef ||| TABLE (External_Coef) — child dataset of External_Coef format. One record per model coefficient.

FIELD response_field ||| STRING — the name of the classifier field (i.e. dependent field name).

SEE External_Coef

RECORD Raw_Prediction

Types \

Raw_Prediction

Record for raw prediction without confidence information.

FIELD raw ||| REAL8 — the raw prediction value.

FIELD number ||| UNSIGNED4 — No Doc

FIELD id ||| UNSIGNED8 — No Doc

FIELD wi ||| UNSIGNED2 — No Doc

RECORD Observation_Deviance

Types \

Observation_Deviance

Record to contain deviance information about each observation.

FIELD nil ||| — ll log likelihood of the nil model (i.e. model with only a constant term).

FIELD mod_ll ||| REAL8 — log likelihood of the model

FIELD classifier ||| UNSIGNED4 — the dependent field number.

FIELD actual ||| INTEGER4 — the actual (i.e. ground truth value).

FIELD nil_dev_residual ||| REAL8 — the deviance not explained by the null model (i.e. the residual)

FIELD wi ||| UNSIGNED2 — the work-item number.

FIELD predicted ||| INTEGER4 — the value predicted by the model.

FIELD mod_dev_residual ||| REAL8 — the deviance not explained by the model (i.e. the residual)

FIELD mod_dev_component ||| REAL8 — the deviance explained by the model

FIELD nil_dev_component ||| REAL8 — the deviance explained by the null model

FIELD id ||| UNSIGNED8 — the record id (i.e. observation number).

FIELD nil_ll ||| REAL8 — No Doc

RECORD Deviance_Record

Types \

Deviance_Record

Record to hold deviance summary information about a model.

FIELD AIC ||| REAL8 — the Akaike Information Criteria value.

FIELD classifier ||| UNSIGNED4 — the classifier number (i.e. field number of the dependent variable).

FIELD df ||| UNSIGNED8 — degrees-of-freedom of the chi squared distribution.

FIELD deviance ||| REAL8 — the total deviance for this classifier.

FIELD wi ||| UNSIGNED2 — the work-item number

RECORD AOD_Record

Types \

AOD_Record

Record to hold Analysis of Deviance (AOD) information for a model.

FIELD classifier ||| UNSIGNED4 — the classifier number (i.e. field number of the dependent variable).

FIELD deviance ||| REAL8 — the total deviance.

FIELD residual_dev ||| REAL8 — the deviance not explained by the model.

FIELD wi ||| UNSIGNED2 — the work-item number

FIELD p ||| — value the probability that the null hypothesis is correct.

FIELD df ||| UNSIGNED8 — degrees of freedom of the chi squared distribution.

FIELD residual_df ||| UNSIGNED8 — No Doc

FIELD p_value ||| REAL8 — No Doc

RECORD FieldName_Mapping

Types \

FieldName_Mapping

Layout used to hold the mapping between a field's number and its name.

FIELD assigned_name ||| STRING — the textual name of the field (e.g. 'age').

FIELD orig_name ||| STRING — typically the field number as a text string (e.g. '2').

RECORD WorkItem_Mapping

Types \

WorkItem_Mapping

Layout used to hold the mapping between a work-item number and a textual name for that work-item.

FIELD orig_wi ||| STRING — the work-item name.

FIELD wi ||| UNSIGNED2 — the work-item number.

RECORD LUCI_Rec

Types \

LUCI_Rec

Layout to store the lines of a generated LUCI model file.

FIELD line ||| STRING — the text for a single line for the LUCI file.

RECORD LUCI_Model_Rqst

Types \

LUCI_Model_Rqst

Format for information to guide the generation of a LUCI file.

FIELD model_name ||| STRING — an expanded name for the model as used in the LUCI L1MD format.

FIELD model_id ||| STRING — a short textual name for the model as used in the LUCI L1MD format.

FIELD wi_list ||| SET (STRING) — can be set to ['ALL'], or can be a list of work-item names.

FIELD score_card_name ||| STRING — the score card name pattern (see LUCI_Model.ecl for details).

FIELD response_field ||| STRING — name of the dependent field (aka classifier name).
