

root

[Go Up](#)

Name	GLM
Version	3.0.0
Description	Generalized Linear Model implementation
License	http://www.apache.org/licenses/LICENSE-2.0
Copyright	Copyright (C) 2017 HPCC Systems
Authors	HPCCSystems
DependsOn	ML_Core-3.1.1, PBblas
Platform	6.2.0

Table of Contents

Apply2CellsBinary.ecl
Iterate matrix and apply function to each pair of cells
BinomialConfusion.ecl
Calculate the Binomial confusion matrix
Confusion.ecl
Generate the confusion matrix, to compare actual versus predicted response variable values
Constants.ecl
Constants used by the GLM bundle
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
Family.ecl Definitions of supported families of Linear Models
GLM.ecl Main GLM regression module
LogitPredict.ecl Predict the category values with 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
Predict.ecl Calculate the score using the appropriate mean function and the the supplied beta coefficients
Types.ecl Type definitions for GLM bundle

Apply2CellsBinary

[Go Up](#)

IMPORTS

std.blas | std.BLAS.Types |

DESCRIPTIONS

FUNCTION Apply2CellsBinary

<code>Types.matrix_t</code>	<code>Apply2CellsBinary</code>
<code>(Types.dimension_t m = 1, Types.dimension_t n = 1, Types.matrix_t x = [], Types.matrix_t y = [], ICellFuncBinary f = ICellFuncBinary)</code>	

Iterate matrix and apply function to each pair of cells.

PARAMETER `y` ||| SET (REAL8) — matrix

PARAMETER `x` ||| SET (REAL8) — matrix

PARAMETER `f` ||| FUNCTION [REAL8 , REAL8 , UNSIGNED4 , UNSIGNED4] (REAL8) —
function to apply

PARAMETER `m` ||| UNSIGNED4 — number of rows

PARAMETER `n` ||| UNSIGNED4 — number of columns

RETURN SET (REAL8) — updated matrix

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.

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

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

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 the GLM bundle. 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_mse](#) : No Documentation Found
9. [id_dispersion](#) : 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_mse

[Constants](#) \

id_mse

No Documentation Found

RETURN INTEGERS —

ATTRIBUTE id_dispersion

[Constants](#) \

id_dispersion

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

ML_Core.Types | Types | Constants | Family |

DESCRIPTIONS

FUNCTION DataStats

<code>DATASET(Types.Data_Info)</code>	DataStats
<code>(DATASET(Core_Types.NumericField) indep, DATASET(Core_Types.NumericField) dep, BOOLEAN dep_details=TRUE, BOOLEAN ind_details=FALSE, Family.FamilyInterface fam=Family.Gaussian)</code>	

Produce summary information about the datasets.

When `dep_details` or `ind_details = FALSE`, indicates the range for the x or y (independent or dependent) columns.

When `dep_details` or `ind_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 `dep` ||| TABLE (NumericField) — data set of dependent variables.

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

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

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

PARAMETER fam ||| INTERFACE (FamilyInterface) — No Doc

PARAMETER ind_details ||| BOOLEAN — No Doc

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 model , UNSIGNED8 residual_df , REAL8 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](#) | [IRLS](#) | [Family](#) | [Constants](#) |

DESCRIPTIONS

FUNCTION `Deviance_Detail`

<code>DATASET(Types.Observation_Deviance)</code>	<code>Deviance_Detail</code>
<code>(DATASET(NumericField) dependents, DATASET(NumericField) predicts, DATASET(Layout_Model) model, Family.FamilyInterface fam)</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).

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

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

PARAMETER `fam` ||| INTERFACE (FamilyInterface) — a module defining the error distribution and link of the dependents

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

RETURN TABLE ({ UNSIGNED2 wi , UNSIGNED8 id , UNSIGNED4 model , REAL8 actual , REAL8 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 C ||| SET (REAL8) — matrix C or empty.

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

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

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

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

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

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

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

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

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

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

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

SEE Std.BLAS.Types.matrix_t

enum__workitems

[Go Up](#)

DESCRIPTIONS

MACRO enum__workitems

enum__workitems
(dsIn, dsOut, src_field, wi_name)

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

PARAMETER dsIn ||| INTEGER8 — the input recordset.

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

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

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

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

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` , 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 (Regressor_Stats) stats }) — the model report in Model_Report format.

SEE Types.Model_Report

Family

[Go Up](#)

IMPORTS

ML_Core |

DESCRIPTIONS

MODULE Family

Family

Definitions of supported families of Linear Models.

Currently supported families are:

- Binomial
- Quasibinomial
- Poisson
- Quasipoisson
- Gamma
- Gaussian
- InvGaussian

In addition, FamilyInterface defines the interface needed to add new families.

Adding new families is fairly straightforward and involves overlaying a set of scalar functions that define the computations for that family. See `FamilyInterface` below.

Children

1. [FamilyInterface](#) : Defines the interface to create new GLM Regression Families
2. [Binomial](#) : The Binomial Regression Family models the response (dependent variable(s)) as a series of Bernoulli Trials, of one of two disjoint outcomes
3. [Quasibinomial](#) : The Quasibinomial Regression Family is similar to the Binomial family (see Binomial above) except that it adjusts for situations where the variance of the distribution is greater or less than anticipated by the Binomial Distribution
4. [Poisson](#) : Poisson Regression is generally used to model count data, where the dependent variable is a positive (or zero) integer
5. [Quasipoisson](#) : Quasipoisson Regression is similar to Poisson Regression (see Poisson above) except that it adjusts for situations where the variance of the distribution is greater or less than anticipated by the Poisson Distribution
6. [Gamma](#) : Gamma Regression is used to model continuous, non-negative, data with a right-skew
7. [Gaussian](#) : Gaussian Regression is equivalent to Ordinary Least Squares (OLS) regression
8. [InvGauss](#) : Inverse Gaussian Regression aka Wald Regression is similar to the Gamma Regression in that it is used to model continuous, positive heteroskedastic data

INTERFACE `FamilyInterface`

[Family](#) \

FamilyInterface

Defines the interface to create new GLM Regression Families. Each family defines a series of eleven attributes that describe the computations for that family within the overall GLM model.

Children

1. [link](#) : This function defines the linkage between output of the linear function on independent data and the dependent data
2. [mu](#) : The Mean function is the inverse of the link function
3. [deta](#) : The derivative of the output of the linear function with respect to the expected value of the dependent variable

4. `var` : The variance as a function of the output value
 5. `init` : Initialization transform sets the initial value for Betas when running Iteratively Re-weighted Least Squares (IRLS)
 6. `ll` : Log Likelihood function
 7. `mu_LUCI` : The string representation of the mu function (see mu above) for use in LUCI
 8. `dispersion` : Flag indicating whether the error distribution should be adjusted for over-dispersion or under-dispersion
 9. `cardinality` : The minimum and maximum cardinality (i.e
 10. `values` : The range of values that the dependent data can take
 11. `isInteger` : Flag that indicates that the dependent variables can only take Integer values
-

FUNCTION `link`

`Family` \ `FamilyInterface` \

<code>REAL8</code>	<code>link</code>
<code>(REAL8 m)</code>	

This function defines the linkage between output of the linear function on independent data and the dependent data.

PARAMETER `m` ||| `REAL8` — The output from the linear function (i.e. the mean)

RETURN `REAL8` — The value to be compared to the dependent data.

FUNCTION `mu`

`Family` \ `FamilyInterface` \

<code>REAL8</code>	<code>mu</code>
<code>(REAL8 v)</code>	

The Mean function is the inverse of the link function. It maps the expected value of the dependent variable to the expected linear result.

PARAMETER \underline{y} ||| REAL8 — The expected value of the dependent variable.

RETURN REAL8 — The expected output from the linear function.

FUNCTION deta

Family \ FamilyInterface \

REAL8	deta
(REAL8 m)	

The derivative of the output of the linear function with respect to the expected value of the dependent variable.

PARAMETER \underline{m} ||| REAL8 — The value of the output.

RETURN REAL8 — The derivative at m.

FUNCTION var

Family \ FamilyInterface \

REAL8	var
(REAL8 m)	

The variance as a function of the output value. This is used for heteroskedastic distributions, otherwise 1.

PARAMETER \underline{m} ||| REAL8 — The value of the output.

RETURN REAL8 — The expected variance when output is at m

FUNCTION `init`

Family \ FamilyInterface \

REAL8	init
(REAL8 y , REAL8 w)	

Initialization transform sets the initial value for Betas when running Iteratively Re-weighted Least Squares (IRLS).

PARAMETER \underline{y} ||| REAL8 — the dependent value.

PARAMETER \underline{w} ||| REAL8 — the current weight.

RETURN REAL8 — the initial weight value to use.

FUNCTION `ll`

Family \ FamilyInterface \

REAL8	ll
(REAL8 y , // log-likelihood function REAL8 m , REAL8 $disp$)	

Log Likelihood function.

PARAMETER \underline{disp} ||| REAL8 — The dispersion factor

PARAMETER \underline{y} ||| REAL8 — The dependent variable.

PARAMETER \underline{m} ||| REAL8 — The output value.

RETURN REAL8 — The log likelihood of seeing m given y .

ATTRIBUTE mu_LUCI

Family \ FamilyInterface \

STRING	mu_LUCI
---------------	----------------

The string representation of the mu function (see mu above) for use in LUCI. See LUCI guide for formatting of this ECL string.

RETURN STRING —

RETURNS An ECL string representation of the mu function.

ATTRIBUTE dispersion

Family \ FamilyInterface \

BOOLEAN	dispersion
----------------	-------------------

Flag indicating whether the error distribution should be adjusted for over-dispersion or under-dispersion.

RETURN BOOLEAN —

ATTRIBUTE cardinality

Family \ FamilyInterface \

SET OF UNSIGNED4	cardinality
-------------------------	--------------------

The minimum and maximum cardinality (i.e. number of unique values) for dependent data.

RETURN SET (UNSIGNED4) — SET([min_cardinality, max_cardinality])

ATTRIBUTE values

Family \ FamilyInterface \

SET OF REALS	values
--------------	--------

The range of values that the dependent data can take.

RETURN SET (REALS) — SET([min_value, max_value])

ATTRIBUTE isInteger

Family \ FamilyInterface \

BOOLEAN	isInteger
---------	-----------

Flag that indicates that the dependent variables can only take Integer values. If FALSE, then REAL values are supported.

RETURN BOOLEAN — Boolean indicating if output is restricted to Integer values.

MODULE Binomial

Family \

Binomial

The Binomial Regression Family models the response (dependent variable(s)) as a series of Bernoulli Trials, of one of two disjoint outcomes.

It is appropriate for modeling a binary result such as success / fail or true / false, which is typical in binary classification problems.

PARENT Family.FamilyInterface <Family.ecl.tex>

MODULE Quasibinomial

[Family \](#)

Quasibinomial

The Quasibinomial Regression Family is similar to the Binomial family (see Binomial above) except that it adjusts for situations where the variance of the distribution is greater or less than anticipated by the Binomial Distribution. This is known as over-dispersion or under-dispersion.

The results are adjusted based on the dispersion of the data to better model the observations in these situations.

PARENT `Family.FamilyInterface` <Family.ecl.tex>

MODULE Poisson

[Family \](#)

Poisson

Poisson Regression is generally used to model count data, where the dependent variable is a positive (or zero) integer.

It is also known as a log-linear model in that the logarithm of the dependent variables is assumed to be linear.

PARENT `Family.FamilyInterface` <Family.ecl.tex>

MODULE Quasipoisson

[Family \](#)

Quasipoisson

Quasipoisson Regression is similar to Poisson Regression (see Poisson above) except that it adjusts for situations where the variance of the distribution is greater or less than anticipated by the Poisson Distribution. This is known as over-dispersion or under-dispersion.

The results are adjusted based on the dispersion of the data to better model the observations in these situations.

PARENT Family.FamilyInterface <Family.ecl.tex>

MODULE Gamma

Family \

Gamma

Gamma Regression is used to model continuous, non-negative, data with a right-skew. Such data exhibits heteroskedasticity, (i.e. inconsistent variance across the range). The gamma regression assumes that the variance is near constant on a log scale. Various types of financial and insurance data often have these characteristics.

PARENT Family.FamilyInterface <Family.ecl.tex>

MODULE Gaussian

Family \

Gaussian

Gaussian Regression is equivalent to Ordinary Least Squares (OLS) regression. It assumes that the error term is Normally distributed.

PARENT Family.FamilyInterface <Family.ecl.tex>

MODULE InvGauss

Family \

InvGauss

Inverse Gaussian Regression aka Wald Regression is similar to the Gamma Regression in that it is used to model continuous, positive heteroskedastic data. It differs from the Gamma Regression assumptions in that it has a wider tail (i.e. more frequent occurrence of higher numbers). The variance is assumed to be proportional to the cube of the mean.

PARENT Family.FamilyInterface <Family.ecl.tex>

GLM

[Go Up](#)

IMPORTS

[Constants](#) | [IRLS](#) | [Family](#) | [ML_Core.Interfaces](#) | [ML_Core.Types](#) |

DESCRIPTIONS

MODULE GLM

GLM
<pre>(DATASET(NumericField) X = DATASET([], NumericField), DATASET(NumericField) Y = DATASET([], NumericField), Family.FamilyInterface fam = Family.Gaussian, DATASET(NumericField) weights = DATASET([], NumericField), UNSIGNED max_iter = 200, REAL8 epsilon = Constants.default_epsilon, REAL8 ridge = Constants.default_ridge)</pre>

Main GLM regression module. Performs regressions using iteratively re-weighted least squares (IRLS).

PARAMETER fam ||| INTERFACE (FamilyInterface) — (Optional) A module defining the type of regression to perform. Default = Gaussian (i.e. ordinary least squares).

PARAMETER weights ||| TABLE (NumericField) — (Optional) A set of observation weights (one per dependent value), in NumericField format. Default = equal weights.

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

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

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

PARAMETER **Y** ||| TABLE (NumericField) — The observed values the model aims to fit in NumericField format.

PARAMETER **X** ||| TABLE (NumericField) — The observed explanatory values in NumericField format.

SEE ML_Core.Types.NumericField

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

Children

1. [GetModel](#) : Calculate a model to fit the observation data to the observed values
2. [Predict](#) : Predict the observations using models trained by the GetModel function

ATTRIBUTE GetModel

GLM \

<code>DATASET(Types.Layout_Model)</code>	GetModel
--	----------

Calculate a model to fit the observation data to the observed values.

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

OVERRIDE

SEE ML_Core.Types.Layout_Model

FUNCTION Predict

GLM \

<code>DATASET(NumericField)</code>	Predict
<code>(DATASET(NumericField) newX, DATASET(Layout_Model) model)</code>	

Predict the observations using models trained by the GetModel function.

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

PARAMETER `newX` ||| TABLE (NumericField) — Observations to be predicted.

RETURN TABLE ({ UNSIGNED2 `wi` , UNSIGNED8 `id` , UNSIGNED4 `number` , REAL8 `value` }) — Predictions in NumericField format.

OVERRIDE

SEE ML_Core.Types.NumericField

LogitPredict

[Go Up](#)

IMPORTS

ML_Core.Types | Types | Family |

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

LUCI_Model

[Go Up](#)

IMPORTS

Types | IRLS | Family | std.Str | std.system.ThorLib |

DESCRIPTIONS

FUNCTION LUCI_Model

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

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 **fam** ||| INTERFACE (FamilyInterface) — the family module for the distribution family on which the regression is based.

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 Family

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

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

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

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

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 `model` , REAL8 `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



Predict

[Go Up](#)

IMPORTS

ML_Core.Types | Types | IRLS | Family |

DESCRIPTIONS

FUNCTION Predict

<code>DATASET(NumericField)</code>	Predict
<code>(DATASET(Model_Coef) coef, DATASET(NumericField) independents, Family.FamilyInterface fam)</code>	

Calculate the score using the appropriate mean function and the the supplied beta coefficients.

PARAMETER `fam` ||| INTERFACE (FamilyInterface) — module defining the error distribution and link of the dependents.

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

PARAMETER `coef` ||| TABLE (Model_Coef) — the model beta coefficients.

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

Types

[Go Up](#)

IMPORTS

ML_Core.Types |

DESCRIPTIONS

MODULE Types

Types

Type definitions for GLM 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. [Data_Diagnostic](#) : Describes any errors in the data
 14. [NumericField_U](#) : Record structure to add a 'Universe Number' to a NumericField allowing multiple independent NumericField matrixes within a work-item
 15. [DiscreteField_U](#) : Record structure to add a 'Universe Number' to a DiscreteField allowing multiple independent DiscreteField matrixes within a work-item
 16. [Layout_Column_Map](#) : Layout for a column map record that is used to remap column numbers
 17. [Regressor_Stats](#) : Summary information about a regressor
 18. [Model_Report](#) : Statistical information about a model
 19. [Binomial_Confusion_Summary](#) : Accuracy stats for binomial classifications
 20. [Model_Coef](#) : Model Coefficients
 21. [Confidence_Model_Coef](#) : Model Coefficients with confidence intervals
 22. [pval_Model_Coef](#) : Model coefficients with z and p-value
 23. [Full_Model_Coef](#) : Model coefficients with confidence intervals and p-value
 24. [External_Coef](#) : Model coefficients, confidence intervals, and p-value, plus independent field names, for each coefficient
 25. [External_Model](#) : Expanded version of a model with statistics and field names
 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 wi ||| UNSIGNED2 — No Doc

FIELD id ||| UNSIGNED8 — No Doc

FIELD number ||| UNSIGNED4 — 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 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.

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

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

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 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.

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

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

RECORD Layout_Model

Types \

Layout_Model

No Documentation Found

FIELD wi ||| UNSIGNED2 — No Doc

FIELD value ||| REAL8 — No Doc

FIELD id ||| UNSIGNED8 — No Doc

FIELD number ||| UNSIGNED4 — 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 max_value ||| REAL8 — the maximum value for the field.

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

FIELD is_integer ||| BOOLEAN — No Doc

RECORD Data_Info

Types \

Data_Info

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

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

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

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

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

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

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

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

FIELD dependent_count ||| UNSIGNED4 — No Doc

FIELD independent_count ||| UNSIGNED4 — No Doc

SEE Field_Desc

RECORD Data_Diagnostic

Types \

Data_Diagnostic

Describes any errors in the data.

FIELD valid ||| BOOLEAN — Boolean TRUE indicates that the data is valid, FALSE indicates problems with the data.

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

FIELD message_text ||| SET (VARSTRING) — A textual description of any errors in the data.

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 wi ||| UNSIGNED2 — No Doc

FIELD value ||| REAL8 — No Doc

FIELD id ||| UNSIGNED8 — No Doc

FIELD number ||| UNSIGNED4 — 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 wi ||| UNSIGNED2 — No Doc

FIELD value ||| INTEGER4 — No Doc

FIELD id ||| UNSIGNED8 — No Doc

FIELD number ||| UNSIGNED4 — No Doc

RECORD Layout_Column_Map

Types \

Layout_Column_Map

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

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

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

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

RECORD Regressor_Stats

Types \

Regressor_Stats

Summary information about a regressor.

FIELD mse ||| REAL8 — the mean square error of the regressor.

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

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

FIELD dispersion ||| REAL8 — the dispersion of the regressor.

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

RECORD Model_Report

Types \

Model_Report

Statistical information about a model.

One record is generated per work-item.

FIELD wi ||| UNSIGNED2 — the work-item

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

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

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

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

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

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

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

SEE Regressor_Stats

RECORD Binomial_Confusion_Summary

Types \

Binomial_Confusion_Summary

Accuracy stats for binomial classifications.

One record per work-item and classifier.

FIELD true_positive ||| UNSIGNED8 — the count of true positive results (i.e. predicted = TRUE, actual = TRUE).

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

FIELD pred_pos ||| UNSIGNED8 — the count of results where predicted = TRUE.

FIELD false_pos_rate ||| REAL8 — false_positive / cond_neg.

FIELD cond_pos ||| UNSIGNED8 — the count of results where actual = TRUE.

FIELD true_negative ||| UNSIGNED8 — the count of true negative results (i.e. predicted = FALSE, actual = FALSE).

FIELD false_positive ||| UNSIGNED8 — the count of false_positive results (i.e. predicted = TRUE, actual = FALSE).

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

FIELD accuracy ||| REAL8 — (true_positive + true_negative) / total.

FIELD true_pos_rate ||| REAL8 — true_positive / cond_pos.

FIELD pos_pred_val ||| REAL8 — true_positive / pred_pos.

FIELD pred_neg ||| UNSIGNED8 — the count of results where predicted = FALSE.

FIELD false_disc_rate ||| REAL8 — false_positive / pred_pos.

FIELD cond_neg ||| UNSIGNED8 — the count of results where actual = FALSE.

FIELD prevalence ||| REAL8 — $\text{cond_pos} / \text{total}$.

FIELD false_omit_rate ||| REAL8 — $\text{false_negative} / \text{pred_neg}$.

FIELD true_neg_rate ||| REAL8 — $\text{true_negative} / \text{cond_neg}$.

FIELD neg_pred_val ||| REAL8 — $\text{true_negative} / \text{pred_neg}$.

FIELD false_negative ||| UNSIGNED8 — the count of false_negative results (i.e. predicted = FALSE, actual = TRUE).

FIELD false_neg_rate ||| REAL8 — No Doc

RECORD Model_Coef

Types \

Model_Coef

Model Coefficients.

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

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

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

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

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

RECORD Confidence_Model_Coef

Types \

Confidence_Model_Coef

Model Coefficients with confidence intervals.

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

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

FIELD se ||| REAL8 — No Doc

FIELD wi ||| UNSIGNED2 — No Doc

FIELD w ||| REAL8 — No Doc

FIELD ind_col ||| UNSIGNED4 — No Doc

FIELD dep_nom ||| UNSIGNED4 — No Doc

RECORD pval_Model_Coef

Types \

pval_Model_Coef

Model coefficients with z and p-value.

FIELD z ||| REAL8 — the z value.

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

FIELD se ||| REAL8 — No Doc

FIELD wi ||| UNSIGNED2 — No Doc

FIELD w ||| REAL8 — No Doc

FIELD ind_col ||| UNSIGNED4 — No Doc

FIELD dep_nom ||| UNSIGNED4 — No Doc

RECORD Full_Model_Coef

Types \

Full_Model_Coef

Model coefficients with confidence intervals and p-value

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

FIELD z ||| REAL8 — the z value.

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

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

FIELD se ||| REAL8 — No Doc

FIELD wi ||| UNSIGNED2 — No Doc

FIELD w ||| REAL8 — No Doc

FIELD ind_col ||| UNSIGNED4 — No Doc

FIELD dep_nom ||| UNSIGNED4 — No Doc

RECORD External_Coef

Types \

External_Coef

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

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

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

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

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

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

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

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

FIELD z ||| REAL8 — the z value.

FIELD ind_col ||| UNSIGNED4 — the field number of the independent field for this 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 response_field ||| STRING — the name of the classifier field (i.e. dependent field name).

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

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

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

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

SEE External_Coef

RECORD Observation_Deviance

Types \

Observation_Deviance

Record to contain deviance information about each observation.

- FIELD** nil_dev_residual ||| REAL8 — the deviance not explained by the null model (i.e. the residual)
 - FIELD** mod_ll ||| REAL8 — log likelihood of the model
 - FIELD** actual ||| REAL8 — the actual (i.e. ground truth value).
 - FIELD** predicted ||| REAL8 — the value predicted by the model.
 - FIELD** nil ||| — ll log likelihood of the nil model (i.e. model with only a constant term).
 - FIELD** mod_dev_residual ||| REAL8 — the deviance not explained by the model (i.e. the residual)
 - FIELD** id ||| UNSIGNED8 — the record id (i.e. observation number).
 - FIELD** mod_dev_component ||| REAL8 — the deviance explained by the model
 - FIELD** classifier ||| — the dependent field number.
 - FIELD** wi ||| UNSIGNED2 — the work-item number.
 - FIELD** nil_dev_component ||| REAL8 — the deviance explained by the null model
 - FIELD** model ||| UNSIGNED4 — No Doc
 - FIELD** nil_ll ||| REAL8 — No Doc
-

RECORD Deviance_Record

Types \

	Deviance_Record
--	-----------------

Record to hold deviance summary information about a model.

- FIELD** df ||| REAL8 — degrees-of-freedom of the chi squared distribution.
 - FIELD** wi ||| UNSIGNED2 — the work-item number
 - FIELD** AIC ||| REAL8 — the Akaike Information Criteria value.
 - FIELD** classifier ||| — the classifier number (i.e. field number of the dependent variable).
 - FIELD** deviance ||| REAL8 — the total deviance for this classifier.
 - FIELD** model ||| UNSIGNED4 — No Doc
-

RECORD AOD_Record

Types \

AOD_Record

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

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

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

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

FIELD df ||| REAL8 — degrees of freedom of the chi squared distribution.

FIELD deviance ||| REAL8 — the total deviance.

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

FIELD model ||| UNSIGNED4 — No Doc

FIELD p_value ||| REAL8 — No Doc

FIELD residual_df ||| UNSIGNED8 — 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 score_card_name ||| STRING — the score card name pattern (see LUCI_Model.ecl for details).

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

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