

root

[Go Up](#)

Name	LogisticRegression
Version	1.0.0
Description	Logistic Regression implementation
License	http://www.apache.org/licenses/LICENSE-2.0
Copyright	Copyright (C) 2017 HPCC Systems
Authors	HPCCSystems
DependsOn	ML_Core, PBblas
Platform	6.2.0

Table of Contents

BinomialConfusion.ecl
Binomial confusion matrix
BinomialLogisticRegression.ecl
Binomial logistic regression using iteratively re-weighted least squares
Confusion.ecl
Detail confusion records to compare actual versus predicted response variable values
Constants.ecl
DataStats.ecl
Information about the datasets
Deviance_Analysis.ecl
Compare deviance information for an analysis of deviance
Deviance_Detail.ecl
Detail deviance for each observation
dimm.ecl
Matrix multiply when either A or B is a diagonal and is passed as a vector
Distributions.ecl
ExtractBeta.ecl

ExtractBeta_CI.ecl
Extract the beta values form the model dataset
ExtractBeta_pval.ecl
Extract the beta values form the model dataset
ExtractReport.ecl
Extract Report records from 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
Model_Deviance.ecl
Model Deviance
Null_Deviance.ecl
Deviance for the null model, that is, a model with only an intercept
Types.ecl
IRLS
performance
Tests
validation

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>	

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` confusion detail for the work item and classifier

RETURN confusion matrix for a binomial classifier

BinomialLogisticRegression

[Go Up](#)

IMPORTS

Constants | ML_Core.Interfaces | ML_Core.Types |

DESCRIPTIONS

MODULE BinomialLogisticRegression

	BinomialLogisticRegression
<pre>(UNSIGNED max_iter=200, REAL8 epsilon=Constants.default_epsilon, REAL8 ridge=Constants.default_ridge)</pre>	

Binomial logistic regression using iteratively re-weighted least squares.

PARAMETER max_iter maximum number of iterations to try

PARAMETER epsilon the minimum change in the Beta value estimate to continue

PARAMETER ridge a value to populate a diagonal matrix that is added to a matrix help assure that the matrix is invertible.

Children

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

FUNCTION GetModel

BinomialLogisticRegression \

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

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

PARAMETER observations the observed explanatory values

PARAMETER classifications the observed classification used to build the model

RETURN the encoded model

OVERRIDE True

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.

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

PARAMETER new_observations observations to be classified

RETURN Classification with a confidence value

OVERRIDE True

FUNCTION Report

BinomialLogisticRegression \

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

Report the confusion matrix for the classifier and training data.

PARAMETER model the encoded model

PARAMETER observations the explanatory values.

PARAMETER classifications the classifications associated with the observations

RETURN the confusion matrix showing correct and incorrect results

OVERRIDE True

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>	

Detail confusion records to compare actual versus predicted response variable values.

PARAMETER dependents the original response values

PARAMETER predicts the predicted responses

RETURN confusion counts by predicted and actual response values.

Constants

[Go Up](#)

DESCRIPTIONS

MODULE Constants

	Constants
--	-----------

Children

1. [limit_card](#)
2. [default_epsilon](#)
3. [default_ridge](#)
4. [local_cap](#)
5. [id_base](#)
6. [id_iters](#)
7. [id_delta](#)
8. [id_correct](#)
9. [id_incorrect](#)
10. [id_stat_set](#)
11. [id_betas](#)
12. [id_betas_coef](#)
13. [id_betas_SE](#)
14. [base_builder](#)
15. [base_max_iter](#)
16. [base_epsilon](#)

- 17. [base_ind_vars](#)
 - 18. [base_dep_vars](#)
 - 19. [base_obs](#)
 - 20. [builder_irls_local](#)
 - 21. [builder_irls_global](#)
 - 22. [builder_softmax](#)
-

ATTRIBUTE `limit_card`

[Constants](#) \

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

ATTRIBUTE `default_epsilon`

[Constants](#) \

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

ATTRIBUTE `default_ridge`

[Constants](#) \

REAL8	<code>default_ridge</code>
--------------	----------------------------

ATTRIBUTE local_cap

Constants \

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

ATTRIBUTE id_base

Constants \

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

ATTRIBUTE id_iters

Constants \

	id_iters
--	----------

ATTRIBUTE id_delta

Constants \

	id_delta
--	----------

ATTRIBUTE id_correct

Constants \

	id_correct
--	------------

ATTRIBUTE id_incorrect

Constants \

	id_incorrect
--	--------------

ATTRIBUTE id_stat_set

Constants \

	id_stat_set
--	-------------

ATTRIBUTE id_betas

Constants \

	id_betas
--	----------

ATTRIBUTE id_betas_coef

Constants \

	id_betas_coef
--	---------------

ATTRIBUTE id_betas_SE

Constants \

	id_betas_SE
--	-------------

ATTRIBUTE base_builder

Constants \

	base_builder
--	--------------

ATTRIBUTE base_max_iter

Constants \

	base_max_iter
--	---------------

ATTRIBUTE base_epsilon

Constants \

	base_epsilon
--	--------------

ATTRIBUTE base_ind_vars

Constants \

	base_ind_vars
--	---------------

ATTRIBUTE base_dep_vars

Constants \

	base_dep_vars
--	---------------

ATTRIBUTE base_obs

Constants \

	base_obs
--	----------

ATTRIBUTE builder_irls_local

Constants \

	builder_irls_local
--	--------------------

ATTRIBUTE builder_irls_global

Constants \

	builder_irls_global
--	---------------------

ATTRIBUTE builder_softmax

Constants \

	builder_softmax
--	-----------------

DataStats

[Go Up](#)

IMPORTS

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

DESCRIPTIONS

FUNCTION DataStats

<code>DATASET(Core_Types.Data_Info)</code>	DataStats
<pre>(DATASET(Core_Types.NumericField) indep, DATASET(Core_Types.DiscreteField) dep, BOOLEAN field_details=FALSE)</pre>	

Information about the datasets. Without details the range for the x and y (independent and dependent) columns. Note that a column of all zero values cannot be distinguished from a missing column. When details are requested, the cardinality, minimum, and maximum values are returned. A zero cardinality is returned when the field cardinality exceeds the Constants.limit_card value.

PARAMETER indep data set of independent variables

PARAMETER dep data set of dependent variables

PARAMETER field_details Boolean directive to provide field level info

Deviance__Analysis

[Go Up](#)

IMPORTS

Types |

DESCRIPTIONS

FUNCTION Deviance__Analysis

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

Compare deviance information for an analysis of deviance.

PARAMETER proposed the proposed model

PARAMETER base the base model for comparison

RETURN the comparison of the deviance between the models

Deviance__Detail

[Go Up](#)

IMPORTS

ML_Core | ML_Core.Types | Types |

DESCRIPTIONS

FUNCTION Deviance__Detail

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

Detail deviance for each observation.

PARAMETER dependents original dependent records for the model

PARAMETER predicts the predicted values of the response variable

RETURN the deviance information by observation and the log likelihood of the predicted result.

dimmm

[Go Up](#)

IMPORTS

std.blas | std.BLAS.Types |

DESCRIPTIONS

EMBED dimmm

<code>Types.matrix_t</code>	dimmm
<pre>(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=[])</pre>	

Matrix multiply when either A or B is a diagonal and is passed as a vector. $\alpha * \text{op}(A) \text{ op}(B) + \beta * C$ where $\text{op}()$ is transpose

PARAMETER transposeA true when transpose of A is used

PARAMETER transposeB true when transpose of B is used

PARAMETER diagonalA true when A is the diagonal matrix

PARAMETER diagonalB true when B is the diagonal matrix

PARAMETER m number of rows in product

PARAMETER n number of columns in product

PARAMETER k number of columns/rows for the multiplier/multiplicand

PARAMETER alpha scalar used on A

PARAMETER A matrix A

PARAMETER B matrix B

PARAMETER beta scalar for matrix C

PARAMETER C matrix C or empty

Distributions

[Go Up](#)

IMPORTS

ML_Core.Constants | ML_Core.Math |

DESCRIPTIONS

MODULE Distributions

	Distributions
--	---------------

Children

1. [Normal_CDF](#) : Cumulative Distribution of the standard normal distribution, the probability that a normal random variable will be smaller than x standard deviations above or below the mean
 2. [Normal_PPF](#) : Normal Distribution Percentage Point Function
 3. [T_CDF](#) : Students t distribution integral evaluated between negative infinity and x
 4. [T_PPF](#) : Percentage point function for the T distribution
 5. [Chi2_CDF](#) : The cumulative distribution function for the Chi Square distribution
 6. [Chi2_PPF](#) : The Chi Squared PPF function
-

FUNCTION Normal_CDF

[Distributions \](#)

REAL8	Normal_CDF
(REAL8 x)	

Cumulative Distribution of the standard normal distribution, the probability that a normal random variable will be smaller than x standard deviations above or below the mean. Taken from C/C++ Mathematical Algorithms for Scientists and Engineers, n. Shamma, McGraw-Hill, 1995

PARAMETER x the number of standard deviations

FUNCTION Normal_PPF

[Distributions \](#)

REAL8	Normal_PPF
(REAL8 x)	

Normal Distribution Percentage Point Function. Translated from C/C++ Mathematical Algorithms for Scientists and Engineers, N. Shamma, McGraw-Hill, 1995

PARAMETER x probability

FUNCTION T_CDF

[Distributions \](#)

REAL8	T_CDF
(REAL8 x, REAL8 df)	

Students t distribution integral evaluated between negative infinity and x. Translated from NIST SEL DATAPAC Fortran TCDF.f source

PARAMETER x value of the evaluation

PARAMETER df degrees of freedom

FUNCTION T_PPF

[Distributions](#) \

REAL8	T_PPF
(REAL8 x, REAL8 df)	

Percentage point function for the T distribution. Translated from NIST SEL DATAPAC Fortran TPPF.f source

FUNCTION Chi2_CDF

[Distributions](#) \

REAL8	Chi2_CDF
(REAL8 x, REAL8 df)	

The cumulative distribution function for the Chi Square distribution. the CDF for the specified degrees of freedom. Translated from the NIST SEL DATAPAC Fortran subroutine CHSCDF.

FUNCTION Chi2_PPF

[Distributions](#) \

REAL8	Chi2_PPF
(REAL8 x, REAL8 df)	

The Chi Squared PPF function. Translated from the NIST SEL DATAPAC Fortran subroutine CHSPPF.

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 the model dataset

RETURN a beta values as Model Coefficient records, zero as the constant term.

ExtractBeta_CI

[Go Up](#)

IMPORTS

ML_Core.Types | Types |

DESCRIPTIONS

FUNCTION ExtractBeta_CI

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

Extract the beta values form the model dataset.

PARAMETER mod_ds the model dataset

PARAMETER level the significance value for the intervals

RETURN the beta values with confidence intervals term.

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 form the model dataset.

PARAMETER mod_ds the model dataset

RETURN the beta values with p-values as Model Coefficient records, zero as the constant term.

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>	

Extract Report records from model

PARAMETER mod_ds the model dataset

RETURN the model report dataset

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 coef the model beta coefficients

PARAMETER independents the observations

RETURN the predicted category values and a confidence score

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 coef the model beta coefficients

PARAMETER independents the observations

RETURN the raw prediction value

Model_Deviance

[Go Up](#)

IMPORTS

Types |

DESCRIPTIONS

FUNCTION Model_Deviance

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

Model Deviance.

PARAMETER od observation deviance record

PARAMETER mod model co-efficients

RETURN model deviance

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>	

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

PARAMETER od Observation Deviance record set.

RETURN a data set of the null model deviances for each work item and classifier.

Types

[Go Up](#)

IMPORTS

ML_Core.Types |

DESCRIPTIONS

MODULE

 Types

	Types
--	-------

Children

- 1. [t_Universe](#)
- 2. [Field_Desc](#)
- 3. [Data_Info](#)
- 4. [NumericField_U](#)
- 5. [DiscreteField_U](#)
- 6. [Layout_Column_Map](#)
- 7. [Classifier_Stats](#)
- 8. [Model_Report](#)
- 9. [Binomial_Confusion_Summary](#)
- 10. [Model_Coef](#)
- 11. [Confidence_Model_Coef](#)

- 12. [pval_Model_Coef](#)
- 13. [Raw_Prediction](#)
- 14. [Observation_Deviance](#)
- 15. [Deviance_Record](#)
- 16. [AOD_Record](#)

ATTRIBUTE t_Universe

[Types](#) \

t_Universe

RECORD Field_Desc

[Types](#) \

Field_Desc

RECORD Data_Info

[Types](#) \

Data_Info

RECORD NumericField_U

[Types](#) \

	NumericField__U
--	-----------------

RECORD

DiscreteField__U

Types \

	DiscreteField__U
--	------------------

RECORD

Layout__Column_Map

Types \

	Layout__Column_Map
--	--------------------

RECORD

Classifier__Stats

Types \

	Classifier__Stats
--	-------------------

RECORD

Model_Report

Types \

	Model_Report
--	--------------

RECORD Binomial_Confusion_Summary

Types \

	Binomial_Confusion_Summary
--	----------------------------

RECORD Model_Coef

Types \

	Model_Coef
--	------------

RECORD Confidence_Model_Coef

Types \

	Confidence_Model_Coef
--	-----------------------

RECORD pval_Model_Coef

Types \

	pval_Model_Coef
--	-----------------

RECORD Raw_Prediction

Types \

	Raw_Prediction
--	----------------

RECORD Observation_Deviance

Types \

	Observation_Deviance
--	----------------------

RECORD Deviance_Record

Types \

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

RECORD AOD_Record

Types \

	AOD_Record
--	------------

IRLS

[Go Up](#)

Table of Contents

GetModel.ecl
Generate logistic regression model from training data
GetModel_global.ecl
Internal function to determine values for the model coefficients and selected statistics from building the model
GetModel_local.ecl
Internal function to determine values for the model co-efficients and selected stats from building the model

IRLS/ GetModel

[Go Up](#)

IMPORTS

ML_Core | ML_Core.Types | Constants | Types | IRLS |

DESCRIPTIONS

FUNCTION GetModel

<code>DATASET(Layout_Model)</code>	GetModel
<pre>(DATASET(NumericField) independents, DATASET(DiscreteField) dependents, UNSIGNED max_iter=200, REAL8 epsilon=Constants.default_epsilon, REAL8 ridge=Constants.default_ridge)</pre>	

Generate logistic regression model from training data. The size of the inputs is used to determine which work items are processed with purely local operations (the data is moved once as necessary) or with global operations supporting a work item to use multiple nodes.

PARAMETER **independents** the independent values

PARAMETER **dependents** the dependent values.

PARAMETER **max_iter** maximum number of iterations to try

PARAMETER **epsilon** the minimum change in the Beta value estimate to continue

PARAMETER **ridge** a value to populate a diagonal matrix that is added to a matrix help assure that the matrix is invertible.

RETURN coefficient matrix plus model building stats

IRLS/ GetModel__global

[Go Up](#)

IMPORTS

ML_Core | ML_Core.Types | PBblas | PBblas.Types | Constants | Types |

DESCRIPTIONS

FUNCTION GetModel__global

<code>DATASET(Layout_Model)</code>	GetModel__global
<pre>(DATASET(NumericField) independents, DATASET(DiscreteField) dependents, UNSIGNED max_iter=200, REAL8 epsilon=Constants.default_epsilon, REAL8 ridge=Constants.default_ridge)</pre>	

Internal function to determine values for the model coefficients and selected statistics from building the model.

PARAMETER **independents** the independent values

PARAMETER **dependents** the dependent values

PARAMETER **max_iter** maximum number of iterations to try

PARAMETER **epsilon** the minimum change in the Beta value estimate to continue

PARAMETER **ridge** a value to pupulate a diagonal matrix that is added to a matrix help assure that the matrix is invertible.

RETURN coefficient matrix plus model building statistics

IRLS/ GetModel__local

[Go Up](#)

IMPORTS

ML_Core | ML_Core.Types | Constants | Types | IRLS | std | std.blas |

DESCRIPTIONS

FUNCTION GetModel__local

<code>DATASET(Layout_Model)</code>	GetModel__local
<pre>(DATASET(NumericField) independents, DATASET(DiscreteField) dependents, UNSIGNED2 max_iter=200, REAL8 epsilon=Constants.default_epsilon, REAL8 ridge=Constants.default_ridge)</pre>	

Internal function to determine values for the model co-efficients and selected stats from building the model.

PARAMETER **independents** the independent values

PARAMETER **dependents** the dependent values.

PARAMETER **max_iter** maximum number of iterations to try

PARAMETER **epsilon** the minimum change in the Beta value estimate to continue

PARAMETER **ridge** a value to populate a diagonal matrix that is added to a matrix help assure that the matrix is invertible.

RETURN coefficient matrix plus model building stats

performance

[Go Up](#)

Table of Contents

RunBinomial.ecl

performance/

RunBinomial

[Go Up](#)

IMPORTS

ML_Core.Types |

DESCRIPTIONS

ATTRIBUTE RunBinomial

	RunBinomial
--	-------------

Tests

[Go Up](#)

Table of Contents

Check_Dist.ecl

Tests/ Check_Dist

[Go Up](#)

IMPORTS

[Distributions](#) | [ML_Core](#) | [python](#) |

DESCRIPTIONS

ATTRIBUTE Check_Dist

	Check_Dist
--	------------

validation

[Go Up](#)

Table of Contents

BinomialRegression.ecl
discrete_GermanDS.ecl
IrisDS.ecl
unit_test_dimm.ecl

validation/ BinomialRegression

[Go Up](#)

IMPORTS

validation | ML_Core.Types | ML_Core | Types |

DESCRIPTIONS

ATTRIBUTE BinomialRegression

	BinomialRegression
--	--------------------

validation/

discrete_GermanDS

[Go Up](#)

IMPORTS

ML_Core.Types |

DESCRIPTIONS

MODULE discrete_GermanDS

	discrete_GermanDS
--	-------------------

Children

1. [content](#)

ATTRIBUTE content

[discrete_GermanDS \](#)

	content
--	---------

[Go Up](#)

IMPORTS

ML_Core | ML_Core.Types |

DESCRIPTIONS

ATTRIBUTE irisDS

	irisDS
--	--------

validation/
unit__test__dimm

[Go Up](#)

IMPORTS

std.BLAS.Types |

DESCRIPTIONS

ATTRIBUTE unit__test__dimm

	unit__test__dimm
--	------------------