

# root

---

[Go Up](#)

## Table of Contents

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

Extract Report records from model
<a href="#">LogitPredict.ecl</a> Predict the category values with the logit function and the the supplied beta coefficients
<a href="#">LogitScore.ecl</a> Calculate the score using the logit function and the the supplied beta coefficients
<a href="#">Model_Deviance.ecl</a> Model Deviance
<a href="#">Null_Deviance.ecl</a> Deviance for the null model, that is, a model with only an intercept
<a href="#">Types.ecl</a>

# BinomialConfusion

---

[Go Up](#)

## IMPORTS

ML\_Core.Types | Types |

## DESCRIPTIONS

### **FUNCTION** BinomialConfusion

<code>DATASET(Types.Binomial_Confusion_Summary)</code>	<b>BinomialConfusion</b>
<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 \

DATASET(Types.Layout_Model)	GetModel
(DATASET(Types.NumericField) observations, DATASET(Types.DiscreteField) classifications)	

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

**PARAMETER** observations the observed explanatory values

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

**RETURN** the encoded model

**OVERRIDE** True

---

## FUNCTION Classify

BinomialLogisticRegression \

DATASET(Types.Classify_Result)	Classify
(DATASET(Types.Layout_Model) model, DATASET(Types.NumericField) new_observations)	

Classify the observations using a model.

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

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

**RETURN** Classification with a confidence value

**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>	<b>Confusion</b>
<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	limit_card
-----------	------------

---

## ATTRIBUTE **default\_epsilon**

[Constants](#) \

REAL8	default_epsilon
-------	-----------------

---

## ATTRIBUTE **default\_ridge**

[Constants](#) \

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

---

## 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>	<b>dimmm</b>
<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](#) \

<b>REAL8</b>	<b>Normal_CDF</b>
(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](#) \

<b>REAL8</b>	<b>Normal_PPF</b>
(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](#) \

<b>REAL8</b>	<b>T_CDF</b>
(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

	<b>ExtractBeta</b>
	(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>	<b>ExtractBeta_CI</b>
<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>	<b>ExtractReport</b>
<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>	<b>LogitPredict</b>
<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>	<b>LogitScore</b>
<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
--	------------

---