

latex-math Macros

compiled: 2021-12-14

Latex macros like `\frac{#1}{#2}` with arguments are displayed as $\frac{\#1}{\#2}$.

Note that macro declarations may only span a single line to be displayed correctly in the below tables.

Contents

basic-math	2
basic-ml	4
ml-ensembles	8
ml-eval	9
ml-feature-sel	11
ml-gp	12
ml-hpo	13
ml-infotheory	14
ml-interpretable	15
ml-nn	16
ml-survival	17
ml-svm	18
ml-trees	19

basic-math

Macro	Notation	Comment
<code>\N</code>	\mathbb{N}	N, naturals
<code>\Z</code>	\mathbb{Z}	Z, integers
<code>\Q</code>	\mathbb{Q}	Q, rationals
<code>\R</code>	\mathbb{R}	R, reals
<code>\C</code>	\mathbb{C}	C, complex
<code>\continuous</code>	\mathcal{C}	C, space of continuous functions
<code>\M</code>	\mathcal{M}	machine numbers
<code>\epsm</code>	ϵ_m	maximum error
<code>\setzo</code>	$\{0, 1\}$	set 0, 1
<code>\setmp</code>	$\{-1, +1\}$	set -1, 1
<code>\unitint</code>	$[0, 1]$	unit interval
<code>\xt</code>	\tilde{x}	x tilde
<code>\argmax</code>	arg max	argmax
<code>\argmin</code>	arg min	argmin
<code>\argminlim</code>	arg min	argmax with limits
<code>\argmaxlim</code>	arg max	argmin with limits
<code>\sign</code>	sign	sign, signum
<code>\I</code>	\mathbb{I}	I, indicator
<code>\order</code>	\mathcal{O}	O, order
<code>\pd</code>	$\frac{\partial \#1}{\partial \#2}$	partial derivative
<code>\floorlr</code>	$\lfloor \#1 \rfloor$	floor
<code>\ceillr</code>	$\lceil \#1 \rceil$	ceiling
<code>\sumin</code>	$\sum_{i=1}^n$	summation from i=1 to n
<code>\sumim</code>	$\sum_{i=1}^m$	summation from i=1 to m
<code>\sumjn</code>	$\sum_{j=1}^n$	summation from j=1 to p
<code>\sumjp</code>	$\sum_{j=1}^p$	summation from j=1 to p
<code>\sumik</code>	$\sum_{i=1}^k$	summation from i=1 to k
<code>\sumkg</code>	$\sum_{k=1}^g$	summation from k=1 to g
<code>\sumjg</code>	$\sum_{j=1}^g$	summation from j=1 to g
<code>\meanin</code>	$\frac{1}{n} \sum_{i=1}^n$	mean from i=1 to n
<code>\meanim</code>	$\frac{1}{m} \sum_{i=1}^m$	mean from i=1 to n
<code>\meankg</code>	$\frac{1}{g} \sum_{k=1}^g$	mean from k=1 to g

<code>\prodin</code>	$\prod_{i=1}^n$	product from i=1 to n
<code>\prodkg</code>	$\prod_{k=1}^g$	product from k=1 to g
<code>\prodjp</code>	$\prod_{j=1}^p$	product from j=1 to p
<code>\one</code>	1	1, unitvector
<code>\zero</code>	0	0-vector
<code>\id</code>	I	I, identity
<code>\diag</code>	diag	diag, diagonal
<code>\trace</code>	tr	tr, trace
<code>\spn</code>	span	span
<code>\scp</code>	$\langle \#1, \#2 \rangle$	<.,.>, scalarproduct
<code>\mat</code>	(#1)	short pmatrix command
<code>\Amat</code>	A	matrix A
<code>\Deltab</code>	Δ	error term for vectors
<code>\P</code>	P	P, probability
<code>\E</code>	E	E, expectation
<code>\var</code>	Var	Var, variance
<code>\cov</code>	Cov	Cov, covariance
<code>\corr</code>	Corr	Corr, correlation
<code>\normal</code>	\mathcal{N}	N of the normal distribution
<code>\iid</code>	$\overset{i.i.d}{\sim}$	dist with i.i.d superscript
<code>\distas</code>	$\overset{\#1}{\sim}$... is distributed as ...

[Back to contents](#)

basic-ml

Macro	Notation	Comment
\Xspace	\mathcal{X}	X, input space
\Yspace	\mathcal{Y}	Y, output space
\nset	$\{1, \dots, n\}$	set from 1 to n
\pset	$\{1, \dots, p\}$	set from 1 to p
\gset	$\{1, \dots, g\}$	set from 1 to g
\Pxy	\mathbb{P}_{xy}	P _{xy}
\Exy	\mathbb{E}_{xy}	E _{xy} : Expectation over random variables xy
\xv	\mathbf{x}	vector x (bold)
\xtil	$\tilde{\mathbf{x}}$	vector x-tilde (bold)
\yv	\mathbf{y}	vector y (bold)
\xy	(\mathbf{x}, y)	observation (x, y)
\xvec	$(x_1, \dots, x_p)^T$	(x1, ..., xp)
\Xmat	\mathbf{X}	Design matrix
\allDatasets	\mathbb{D}	The set of all datasets
\allDatasetsn	\mathbb{D}_n	The set of all datasets of size n
\D	\mathcal{D}	D, data
\Dn	\mathcal{D}_n	D _n , data of size n
\Dtrain	$\mathcal{D}_{\text{train}}$	D _{train} , training set
\Dtest	$\mathcal{D}_{\text{test}}$	D _{test} , test set
\xyi	$(\mathbf{x}^{(\#1)}, y^{(\#1)})$	(x ^{~i} , y ^{~i}), i-th observation
\Dset	$((\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(n)}, y^{(n)}))$	{(x1,y1)}, ..., (xn,yn)}, data
\defAllDatasetsn	$(\mathcal{X} \times \mathcal{Y})^n$	Def. of the set of all datasets of size n
\defAllDatasets	$\bigcup_{n \in \mathbb{N}} (\mathcal{X} \times \mathcal{Y})^n$	Def. of the set of all datasets
\xdat	$\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}$	{x1, ..., xn}, input data
\yvec	$(y^{(1)}, \dots, y^{(n)})^T$	(y1, ..., yn), vector of outcomes
\xi	$\mathbf{x}^{(\#1)}$	x ^{~i} , i-th observed value of x
\yi	$y^{(\#1)}$	y ^{~i} , i-th observed value of y
\xivec	$(x_1^{(i)}, \dots, x_p^{(i)})^T$	(x1 ^{~i} , ..., xp ^{~i}), i-th observation vector
\xj	\mathbf{x}_j	x _j , j-th feature
\xjvec	$(x_j^{(1)}, \dots, x_j^{(n)})^T$	(x1 ^{~j} , ..., xn ^{~j}), j-th feature vector
\phiv	ϕ	Basis transformation function phi
\phixi	$\phi^{(i)}$	Basis transformation of xi: phi ^{~i} := phi(xi)
\lamv	$\boldsymbol{\lambda}$	lambda vector, hyperconfiguration vector
\Lam	$\boldsymbol{\Lambda}$	Lambda, space of all hpos
\preimageInducer	$(\bigcup_{n \in \mathbb{N}} (\mathcal{X} \times \mathcal{Y})^n) \times \boldsymbol{\Lambda}$	Set of all datasets times the hyperparameter space
\preimageInducerShort	$\mathbb{D} \times \boldsymbol{\Lambda}$	Set of all datasets times the hyperparameter space
\ind	\mathcal{I}	Inducer, inducing algorithm, learning algorithm
\ftrue	f_{true}	True underlying function (if a statistical model is assumed)
\ftruex	$f_{\text{true}}(\mathbf{x})$	True underlying function (if a statistical model is assumed)
\fx	$f(\mathbf{x})$	f(x), continuous prediction function
\fdomains	$f : \mathcal{X} \rightarrow \mathbb{R}^g$	f with domain and co-domain

<code>\Hspace</code>	\mathcal{H}	hypothesis space where f is from
<code>\fbayes</code>	f^*	Bayes-optimal model
<code>\fxbayes</code>	$f^*(\mathbf{x})$	Bayes-optimal model
<code>\fkx</code>	$f_{\#1}(\mathbf{x})$	$f_{_j}(\mathbf{x})$, discriminant component function
<code>\fh</code>	\hat{f}	f hat, estimated prediction function
<code>\fxh</code>	$\hat{f}(\mathbf{x})$	fhat(x)
<code>\fxt</code>	$f(\mathbf{x} \mid \boldsymbol{\theta})$	$f(\mathbf{x} \mid \text{theta})$
<code>\fxi</code>	$f(\mathbf{x}^{(i)})$	$f(\mathbf{x}^{(i)})$
<code>\fxih</code>	$\hat{f}(\mathbf{x}^{(i)})$	$f(\mathbf{x}^{(i)})$
<code>\fxit</code>	$f(\mathbf{x}^{(i)} \mid \boldsymbol{\theta})$	$f(\mathbf{x}^{(i)} \mid \text{theta})$
<code>\fhD</code>	$\hat{f}_{\mathcal{D}}$	fhat_D, estimate of f based on D
<code>\fhDtrain</code>	$\hat{f}_{\mathcal{D}_{\text{train}}}$	fhat_Dtrain, estimate of f based on D
<code>\fhDnlam</code>	$\hat{f}_{\mathcal{D}_n, \lambda}$	model learned on Dn with hp lambda
<code>\fhDlam</code>	$\hat{f}_{\mathcal{D}, \lambda}$	model learned on D with hp lambda
<code>\fhDnlams</code>	$\hat{f}_{\mathcal{D}_n, \lambda^*}$	model learned on Dn with optimal hp lambda
<code>\fhDlams</code>	$\hat{f}_{\mathcal{D}, \lambda^*}$	model learned on D with optimal hp lambda
<code>\hx</code>	$h(\mathbf{x})$	$h(\mathbf{x})$, discrete prediction function
<code>\hh</code>	\hat{h}	h hat
<code>\hxx</code>	$\hat{h}(\mathbf{x})$	hhat(x)
<code>\hxt</code>	$h(\mathbf{x} \mid \boldsymbol{\theta})$	$h(\mathbf{x} \mid \text{theta})$
<code>\hxi</code>	$h(\mathbf{x}^{(i)})$	$h(\mathbf{x}^{(i)})$
<code>\hxit</code>	$h(\mathbf{x}^{(i)} \mid \boldsymbol{\theta})$	$h(\mathbf{x}^{(i)} \mid \text{theta})$
<code>\hbayes</code>	h^*	Bayes-optimal classification model
<code>\hxbayes</code>	$h^*(\mathbf{x})$	Bayes-optimal classification model
<code>\yh</code>	\hat{y}	yhat for prediction of target
<code>\yih</code>	$\hat{y}^{(i)}$	yhat^(i) for prediction of ith targiet
<code>\resi</code>	$y^{(i)} - \hat{y}^{(i)}$	
<code>\thetah</code>	$\hat{\boldsymbol{\theta}}$	theta hat
<code>\thetab</code>	$\boldsymbol{\theta}$	theta vector
<code>\thetabh</code>	$\hat{\boldsymbol{\theta}}$	theta vector hat
<code>\thetat</code>	$\boldsymbol{\theta}^{[\#1]}$	theta^[t] in optimization
<code>\thetatn</code>	$\boldsymbol{\theta}^{[\#1+1]}$	theta^[t+1] in optimization
<code>\thetahDnlam</code>	$\hat{\boldsymbol{\theta}}_{\mathcal{D}_n, \lambda}$	theta learned on Dn with hp lambda
<code>\thetahDlam</code>	$\hat{\boldsymbol{\theta}}_{\mathcal{D}, \lambda}$	theta learned on D with hp lambda
<code>\mint</code>	$\min_{\boldsymbol{\theta} \in \Theta}$	min problem theta
<code>\argmint</code>	$\arg \min_{\boldsymbol{\theta} \in \Theta}$	argmin theta
<code>\pdf</code>	p	p
<code>\pdfx</code>	$p(\mathbf{x})$	$p(\mathbf{x})$
<code>\pixt</code>	$\pi(\mathbf{x} \mid \boldsymbol{\theta})$	$\text{pi}(\mathbf{x} \mid \text{theta})$, pdf of x given theta
<code>\pixit</code>	$\pi(\mathbf{x}^{(i)} \mid \boldsymbol{\theta})$	$\text{pi}(\mathbf{x}^{(i)} \mid \text{theta})$, pdf of x given theta
<code>\pixii</code>	$\pi(\mathbf{x}^{(i)})$	$\text{pi}(\mathbf{x}^{(i)})$, pdf of i-th x
<code>\pdfxy</code>	$p(\mathbf{x}, y)$	$p(\mathbf{x}, y)$
<code>\pdfxyt</code>	$p(\mathbf{x}, y \mid \boldsymbol{\theta})$	$p(\mathbf{x}, y \mid \text{theta})$
<code>\pdfxyit</code>	$p(\mathbf{x}^{(i)}, y^{(i)} \mid \boldsymbol{\theta})$	$p(\mathbf{x}^{(i)}, y^{(i)} \mid \text{theta})$

<code>\pdfxyk</code>	$p(\mathbf{x} y = \#1)$	$p(\mathbf{x} y = k)$
<code>\lpdfxyk</code>	$\log p(\mathbf{x} y = \#1)$	$\log p(\mathbf{x} y = k)$
<code>\pdfxiyk</code>	$p(\mathbf{x}^{(i)} y = \#1)$	$p(\mathbf{x}^i y = k)$
<code>\pik</code>	$\pi_{\#1}$	π_k , prior
<code>\lpik</code>	$\log \pi_{\#1}$	$\log \pi_k$, log of the prior
<code>\pit</code>	$\pi(\boldsymbol{\theta})$	Prior probability of parameter theta
<code>\post</code>	$\mathbb{P}(y = 1 \mathbf{x})$	$P(y = 1 \mathbf{x})$, post. prob for y=1
<code>\postk</code>	$\mathbb{P}(y = \#1 \mathbf{x})$	$P(y = k \mathbf{x})$, post. prob for y=k
<code>\pidomains</code>	$\pi : \mathcal{X} \rightarrow [0, 1]$	pi with domain and co-domain
<code>\pibayes</code>	π^*	Bayes-optimal classification model
<code>\pixbayes</code>	$\pi^*(\mathbf{x})$	Bayes-optimal classification model
<code>\pix</code>	$\pi(\mathbf{x})$	$\pi(\mathbf{x})$, $P(y = 1 \mathbf{x})$
<code>\pikx</code>	$\pi_{\#1}(\mathbf{x})$	$\pi_k(\mathbf{x})$, $P(y = k \mathbf{x})$
<code>\pikxt</code>	$\pi_{\#1}(\mathbf{x} \boldsymbol{\theta})$	$\pi_k(\mathbf{x} \theta)$, $P(y = k \mathbf{x}, \theta)$
<code>\pixh</code>	$\hat{\pi}(\mathbf{x})$	$\pi(\mathbf{x})$ hat, $P(y = 1 \mathbf{x})$ hat
<code>\pikxh</code>	$\hat{\pi}_{\#1}(\mathbf{x})$	$\pi_k(\mathbf{x})$ hat, $P(y = k \mathbf{x})$ hat
<code>\pixih</code>	$\hat{\pi}(\mathbf{x}^{(i)})$	$\pi(\mathbf{x}^i)$ with hat
<code>\pikxih</code>	$\hat{\pi}_{\#1}(\mathbf{x}^{(i)})$	$\pi_k(\mathbf{x}^i)$ with hat
<code>\pdfygxt</code>	$p(y \mathbf{x}, \boldsymbol{\theta})$	$p(y \mathbf{x}, \theta)$
<code>\pdfyigxit</code>	$p(y^{(i)} \mathbf{x}^{(i)}, \boldsymbol{\theta})$	$p(y^i \mathbf{x}^i, \theta)$
<code>\lpdfygxt</code>	$\log p(y \mathbf{x}, \boldsymbol{\theta})$	$\log p(y \mathbf{x}, \theta)$
<code>\lpdfyigxit</code>	$\log p(y^{(i)} \mathbf{x}^{(i)}, \boldsymbol{\theta})$	$\log p(y^i \mathbf{x}^i, \theta)$
<code>\bayesrulek</code>	$\frac{\mathbb{P}(\mathbf{x} y=\#1)\mathbb{P}(y=\#1)}{\mathbb{P}(\mathbf{x})}$	Bayes rule
<code>\muk</code>	$\boldsymbol{\mu}_k$	mean vector of class-k Gaussian (discr analysis)
<code>\eps</code>	ϵ	residual, stochastic
<code>\epsi</code>	$\epsilon^{(i)}$	ϵ^i , residual, stochastic
<code>\epsh</code>	$\hat{\epsilon}$	residual, estimated
<code>\yf</code>	$yf(\mathbf{x})$	$y f(\mathbf{x})$, margin
<code>\yfi</code>	$y^{(i)}f(\mathbf{x}^{(i)})$	$y^i f(\mathbf{x}^i)$, margin
<code>\Sigmah</code>	$\hat{\Sigma}$	estimated covariance matrix
<code>\Sigmahj</code>	$\hat{\Sigma}_j$	estimated covariance matrix for the j-th class
<code>\Lyf</code>	$L(y, f)$	$L(y, f)$, loss function
<code>\Lxy</code>	$L(y, f(\mathbf{x}))$	$L(y, f(\mathbf{x}))$, loss function
<code>\Lxyi</code>	$L(y^{(i)}, f(\mathbf{x}^{(i)}))$	loss of observation
<code>\Lxyt</code>	$L(y, f(\mathbf{x} \boldsymbol{\theta}))$	loss with f parameterized
<code>\Lxyit</code>	$L(y^{(i)}, f(\mathbf{x}^{(i)} \boldsymbol{\theta}))$	loss of observation with f parameterized
<code>\Lxym</code>	$L(y^{(i)}, f(\tilde{\mathbf{x}}^{(i)} \boldsymbol{\theta}))$	loss of observation with f parameterized
<code>\Lpixy</code>	$L(y, \pi(\mathbf{x}))$	loss in classification
<code>\Lpixyi</code>	$L(y^{(i)}, \pi(\mathbf{x}^{(i)}))$	loss of observation in classification
<code>\Lpixyt</code>	$L(y, \pi(\mathbf{x} \boldsymbol{\theta}))$	loss with pi parameterized
<code>\Lpixyit</code>	$L(y^{(i)}, \pi(\mathbf{x}^{(i)} \boldsymbol{\theta}))$	loss of observation with pi parameterized
<code>\Lhxy</code>	$L(y, h(\mathbf{x}))$	$L(y, h(\mathbf{x}))$, loss function on discrete classes
<code>\Lr</code>	$L(r)$	$L(r)$, loss defined on residual (reg) / margin (classif)
<code>\lone</code>	$ y - f(\mathbf{x}) $	L1 loss
<code>\ltwo</code>	$(y - f(\mathbf{x}))^2$	L2 loss

<code>\lbernoullimp</code>	$\ln(1 + \exp(-y \cdot f(\mathbf{x})))$	Bernoulli loss for -1, +1 encoding
<code>\lbernoullizo</code>	$-y \cdot f(\mathbf{x}) + \log(1 + \exp(f(\mathbf{x})))$	Bernoulli loss for 0, 1 encoding
<code>\lcrossent</code>	$-y \log(\pi(\mathbf{x})) - (1 - y) \log(1 - \pi(\mathbf{x}))$	cross-entropy loss
<code>\lbrier</code>	$(\pi(\mathbf{x}) - y)^2$	Brier score
<code>\risk</code>	\mathcal{R}	R, risk
<code>\riskbayes</code>	\mathcal{R}^*	
<code>\riskf</code>	$\mathcal{R}(f)$	R(f), risk
<code>\riskdef</code>	$\mathbb{E}_{y \mathbf{x}}(L(y, f(\mathbf{x})))$	risk def (expected loss)
<code>\riskt</code>	$\mathcal{R}(\boldsymbol{\theta})$	R(theta), risk
<code>\riske</code>	\mathcal{R}_{emp}	R_emp, empirical risk w/o factor 1 / n
<code>\riskeb</code>	$\bar{\mathcal{R}}_{\text{emp}}$	R_emp, empirical risk w/ factor 1 / n
<code>\riskef</code>	$\mathcal{R}_{\text{emp}}(f)$	R_emp(f)
<code>\risket</code>	$\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$	R_emp(theta)
<code>\riskr</code>	\mathcal{R}_{reg}	R_reg, regularized risk
<code>\riskrt</code>	$\mathcal{R}_{\text{reg}}(\boldsymbol{\theta})$	R_reg(theta)
<code>\riskrf</code>	$\mathcal{R}_{\text{reg}}(f)$	R_reg(f)
<code>\riskrth</code>	$\hat{\mathcal{R}}_{\text{reg}}(\boldsymbol{\theta})$	hat R_reg(theta)
<code>\risketh</code>	$\hat{\mathcal{R}}_{\text{emp}}(\boldsymbol{\theta})$	hat R_emp(theta)
<code>\LL</code>	\mathcal{L}	L, likelihood
<code>\LLt</code>	$\mathcal{L}(\boldsymbol{\theta})$	L(theta), likelihood
<code>\LLtx</code>	$\mathcal{L}(\boldsymbol{\theta} \mathbf{x})$	L(theta x), likelihood
<code>\logl</code>	ℓ	l, log-likelihood
<code>\loglt</code>	$\ell(\boldsymbol{\theta})$	l(theta), log-likelihood
<code>\logltx</code>	$\ell(\boldsymbol{\theta} \mathbf{x})$	l(theta x), log-likelihood
<code>\errtrain</code>	$\text{err}_{\text{train}}$	training error
<code>\errtest</code>	err_{test}	test error
<code>\errexp</code>	$\overline{\text{err}_{\text{test}}}$	avg training error
<code>\thx</code>	$\boldsymbol{\theta}^T \mathbf{x}$	linear model
<code>\olsest</code>	$(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$	OLS estimator in LM

[Back to contents](#)

ml-ensembles

Macro	Notation	Comment
\bl	$b^{[\#1]}$	baselearner, default m
\blh	$\hat{b}^{[\#1]}$	estimated base learner, default m
\blx	$b^{[\#1]}(\mathbf{x})$	baselearner, default m
\fM	$f^{[M]}(\mathbf{x})$	ensembled predictor
\fMh	$\hat{f}^{[M]}(\mathbf{x})$	estimated ensembled predictor
\ambifM	$\Delta(f^{[M]}(\mathbf{x}))$	ambiguity/instability of ensemble
\betam	$\beta^{[\#1]}$	weight of basemodel m
\betamh	$\hat{\beta}^{[\#1]}$	weight of basemodel m with hat
\betaM	$\beta^{[M]}$	last baselearner
\fm	$f^{[\#1]}$	prediction in iteration m
\fmh	$\hat{f}^{[\#1]}$	prediction in iteration m
\fmd	$f^{[\#1-1]}$	prediction m-1
\fmdh	$\hat{f}^{[\#1-1]}$	prediction m-1
\errm	$\text{err}^{[\#1]}$	weighted in-sample misclassification rate
\wm	$w^{[\#1]}$	weight vector of basemodel m
\wmi	$w^{[\#1](i)}$	weight of obs i of basemodel m
\thetam	$\boldsymbol{\theta}^{[\#1]}$	parameters of basemodel m
\thetamh	$\hat{\boldsymbol{\theta}}^{[\#1]}$	parameters of basemodel m with hat
\blxt	$b(\mathbf{x}, \boldsymbol{\theta}^{[\#1]})$	baselearner, default m
\ens	$\sum_{m=1}^M \beta^{[m]} b(\mathbf{x}, \boldsymbol{\theta}^{[m]})$	ensemble
\rmm	$\tilde{r}^{[\#1]}$	pseudo residuals
\rmi	$\tilde{r}^{[\#1](i)}$	pseudo residuals
\Rtm	$R_t^{[\#1]}$	terminal-region
\Tm	$T^{[\#1]}$	terminal-region
\ctm	$c_t^{[\#1]}$	mean, terminal-regions
\ctmh	$\hat{c}_t^{[\#1]}$	mean, terminal-regions with hat
\ctmt	$\tilde{c}_t^{[\#1]}$	mean, terminal-regions
\Lp	L'	
\Ldp	L''	
\Lpleft	L'_{left}	

[Back to contents](#)

ml-eval

Macro	Notation	Comment
<code>\ntest</code>	n_{test}	size of the test set
<code>\ntrain</code>	n_{train}	size of the train set
<code>\ntesti</code>	$n_{\text{test},\#1}$	size of the i-th test set
<code>\ntraini</code>	$n_{\text{train},\#1}$	size of the i-th train set
<code>\Jtrain</code>	J_{train}	index vector train data
<code>\Jtest</code>	J_{test}	index vector test data
<code>\Jtraini</code>	$J_{\text{train},\#1}$	index vector i-th train dataset
<code>\Jtesti</code>	$J_{\text{test},\#1}$	index vector i-th test dataset
<code>\Dtraini</code>	$\mathcal{D}_{\text{train},\#1}$	$\mathcal{D}_{\text{train},i}$, i-th training set
<code>\Dtesti</code>	$\mathcal{D}_{\text{test},\#1}$	$\mathcal{D}_{\text{test},i}$, i-th test set
<code>\JSpace</code>	$\{1, \dots, n\}^{\#1}$	space of train indices of size n_{train}
<code>\JtrainSpace</code>	$\{1, \dots, n\}^{n_{\text{train}}}$	space of train indices of size n_{train}
<code>\JtestSpace</code>	$\{1, \dots, n\}^{n_{\text{test}}}$	space of train indices of size n_{test}
<code>\yJ</code>	$\mathbf{y}_{\#1}$	output vector associated to index J
<code>\yJDef</code>	$\left(y^{(J^{(1)})}, \dots, y^{(J^{(m)})}\right)$	def of the output vector associated to index J
<code>\JJ</code>	\mathcal{J}	cali-J, set of all splits
<code>\JJset</code>	$((J_{\text{train},1}, J_{\text{test},1}), \dots, (J_{\text{train},B}, J_{\text{test},B}))$	$(J_{\text{train}_1}, J_{\text{test}_1}) \dots (J_{\text{train}_B}, J_{\text{test}_B})$
<code>\Itrainlam</code>	$\mathcal{I}(\mathcal{D}_{\text{train}}, \boldsymbol{\lambda})$	
<code>\GE</code>	$\widehat{\text{GE}}$	GE
<code>\GEh</code>	$\widehat{\text{GE}}$	GE-hat
<code>\GEfull</code>	$\widehat{\text{GE}}(\mathcal{I}, \boldsymbol{\lambda}, \#1, \rho)$	GE full
<code>\GEholdout</code>	$\widehat{\text{GE}}_{J_{\text{train}}, J_{\text{test}}}(\mathcal{I}, \boldsymbol{\lambda}, J_{\text{train}} , \rho)$	GE hat holdout
<code>\GEholdouti</code>	$\widehat{\text{GE}}_{J_{\text{train},\#1}, J_{\text{test},\#1}}(\mathcal{I}, \boldsymbol{\lambda}, J_{\text{train},\#1} , \rho)$	GE hat holdout i-th set
<code>\GEhlam</code>	$\widehat{\text{GE}}(\boldsymbol{\lambda})$	GE-hat(lam)
<code>\GEhlamsubIJrho</code>	$\widehat{\text{GE}}_{\mathcal{I}, \mathcal{J}, \rho}(\boldsymbol{\lambda})$	GE-hat_I,J,rho(lam)
<code>\GEhresa</code>	$\widehat{\text{GE}}(\mathcal{I}, \mathcal{J}, \rho, \boldsymbol{\lambda})$	GE-hat_I,J,rho(lam)
<code>\GERhoDef</code>	$\lim_{n_{\text{test}} \rightarrow \infty} \mathbb{E}_{\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \sim \mathbb{P}_{xy}} \left[\rho(\mathbf{y}_{J_{\text{test}}}, \mathbf{F}_{J_{\text{test}}, \mathcal{I}(\mathcal{D}_{\text{train}}, \boldsymbol{\lambda})}) \right]$	GE formal def
<code>\agr</code>	agr	aggregate function
<code>\GEf</code>	$\widehat{\text{GE}}\left(\hat{f}\right)$	GE of a fitted model
<code>\GEfh</code>	$\widehat{\text{GE}}\left(\hat{f}\right)$	GEh of a fitted model
<code>\GEfL</code>	$\widehat{\text{GE}}\left(\hat{f}, L\right)$	GE of a fitted model wrt loss L
<code>\Lyfhx</code>	$L\left(y, \hat{f}(\mathbf{x})\right)$	pointwise loss of fitted model
<code>\GEnf</code>	$GE_n\left(\hat{f}_{\#1}\right)$	GE of a fitted model
<code>\GEind</code>	$GE_n(\mathcal{I}_{L,O})$	GE of inducer
<code>\GED</code>	$\text{GE}_{\mathcal{D}}$	GE indexed with data
<code>\EGEn</code>	EGE_n	expected GE
<code>\EDn</code>	$\mathbb{E}_{ D =n}$	expectation wrt data of size n
<code>\rhoL</code>	ρ_L	perf. measure derived from pointwise loss
<code>\F</code>	\mathbf{F}	matrix of prediction scores

<code>\Fi</code>	$\mathbf{F}^{(\#1)}$	i-th row vector of the predsore mat
<code>\FJ</code>	$\mathbf{F}_{\#1}$	predscore mat idxvec J
<code>\FJf</code>	$\mathbf{F}_{J,f}$	predscore mat idxvec J and model f
<code>\FJtestfh</code>	$\mathbf{F}_{J_{\text{test}},\hat{f}}$	predscore mat idxvec Jtest and model f hat
<code>\FJtestftrain</code>	$\mathbf{F}_{J_{\text{test}},\mathcal{I}(\mathcal{D}_{\text{train}},\boldsymbol{\lambda})}$	predscore mat idxvec Jtest and model f
<code>\FJtestftraini</code>	$\mathbf{F}_{J_{\text{test}},\#1,\mathcal{I}(\mathcal{D}_{\text{train}},\#1,\boldsymbol{\lambda})}$	predscore mat i-th idxvec Jtest and model f
<code>\FJfDef</code>	$\left(f(\mathbf{x}^{(J^{(1)})}), \dots, f(\mathbf{x}^{(J^{(m)})})\right)$	def of predsore mat idxvec J and model f
<code>\preimageRho</code>	$\bigcup_{m \in \mathbb{N}} (\mathcal{Y}^m \times \mathbb{R}^{m \times g})$	Set of all datasets times HP space
<code>\np</code>	n_+	no. of positive instances
<code>\nn</code>	n_-	no. of negative instances
<code>\rn</code>	π_-	proportion negative instances
<code>\rp</code>	π_+	proportion negative instances
<code>\tp</code>	$\#TP$	true pos
<code>\fap</code>	$\#FP$	false pos (fp taken for partial derivs)
<code>\tn</code>	$\#TN$	true neg
<code>\fan</code>	$\#FN$	false neg

[Back to contents](#)

ml-feature-sel

Macro	Notation	Comment
<code>\xjNull</code>	x_{j_0}	
<code>\xjEins</code>	x_{j_1}	
<code>\xl</code>	\mathbf{x}_l	
<code>\pushcode</code>		

[Back to contents](#)

ml-gp

Macro	Notation	Comment
<code>\fvec</code>	$[f(\mathbf{x}^{(1)}), \dots, f(\mathbf{x}^{(n)})]$	function vector
<code>\fv</code>	\mathbf{f}	function vector
<code>\kv</code>	\mathbf{k}	cov matrix partition
<code>\kxxp</code>	$k(\mathbf{x}, \mathbf{x}')$	cov of \mathbf{x}, \mathbf{x}'
<code>\kxij</code>	$k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$	cov of $\mathbf{x}_i, \mathbf{x}_j$
<code>\mv</code>	\mathbf{m}	GP mean vector
<code>\Kmat</code>	\mathbf{K}	GP cov matrix
<code>\gaussmk</code>	$\mathcal{N}(\mathbf{m}, \mathbf{K})$	Gaussian w/ mean vec, cov mat
<code>\gp</code>	$\mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$	Gaussian Process Definition
<code>\ls</code>	ℓ	length-scale
<code>\sqexpkernel</code>	$\exp\left(-\frac{\ \mathbf{x}-\mathbf{x}'\ ^2}{2\ell^2}\right)$	squared exponential kernel
<code>\fstarvec</code>	$[f(\mathbf{x}_*^{(1)}), \dots, f(\mathbf{x}_*^{(m)})]$	pred function vector
<code>\kstar</code>	\mathbf{k}_*	cov of new obs with \mathbf{x}
<code>\kstarstar</code>	\mathbf{k}_{**}	cov of new obs
<code>\Kstar</code>	\mathbf{K}_*	cov mat of new obs with \mathbf{x}
<code>\Kstarstar</code>	\mathbf{K}_{**}	cov mat of new obs
<code>\preddistsingle</code>	$f_* \mid \mathbf{x}_*, \mathbf{X}, \mathbf{f}$	predictive distribution for single pred
<code>\preddistdefsingl</code>	$\mathcal{N}(\mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{f}, \mathbf{k}_{**} - \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{k}_*)$	Gaussian distribution for single pred
<code>\preddist</code>	$f_* \mid \mathbf{X}_*, \mathbf{X}, \mathbf{f}$	predictive distribution
<code>\preddistdef</code>	$\mathcal{N}(\mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{f}, \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_*)$	Gaussian predictive distribution

[Back to contents](#)

ml-hpo

Macro	Notation	Comment
\Ilam	\mathcal{I}_{λ}	inducer with HP
\LamS	$\hat{\Lambda}$	search space
\lami	$\lambda^{(\#1)}$	lambda i
\clam	$c(\lambda)$	c(lambda)
\clamh	$c(\hat{\lambda})$	c(lambda-hat)
\lams	λ^*	theoretical min of c
\lamh	$\hat{\lambda}$	returned lambda of HPO
\lamp	λ^+	proposed lambda
\clamp	$c(\lambda^+)$	c of proposed lambda
\archive	\mathcal{A}	archive
\archivet	$\mathcal{A}^{[\#1]}$	archive at time step t
\tuner	\mathcal{T}	tuner
\tunerfull	$\mathcal{T}_{\mathcal{I}, \hat{\Lambda}, \rho, \mathcal{J}}$	tuner with inducer, search space, perf measure, resampling strategy
\chlam	$\hat{c}(\lambda)$	post mean of SM
\shlam	$\hat{\sigma}(\lambda)$	post sd of SM
\vhlam	$\hat{\sigma}^2(\lambda)$	post var of SM
\ulam	$u(\lambda)$	acquisition function
\lambdaopt	λ^*	minimum of the black box function Psi
\metadata	$\{(\lambda^{(i)}, \Psi^{[i]})\}$	metadata for the Gaussian process
\lamvec	$(\lambda^{[1]}, \dots, \lambda^{[m_{\text{init}}]})$	vector of different inputs
\minit	m_{init}	size of the initial design
\lambu	λ_{budget}	single lambda_budget component HP
\lamfid	λ_{fid}	single lambda fidelity
\lamfidl	$\lambda_{\text{fid}}^{\text{low}}$	single lambda fidelity lower
\lamfidu	$\lambda_{\text{fid}}^{\text{upp}}$	single lambda fidelity upper
\etahb	η_{HB}	HB multiplier eta

[Back to contents](#)

ml-infotheory

Macro	Notation	Comment
<code>\entx</code>	$-\sum_{x \in \mathcal{X}} p(x) \cdot \log p(x)$	entropy of x
<code>\dentx</code>	$-\int_{\mathcal{X}} f(x) \cdot \log f(x) dx$	diff entropy of x
<code>\jentyx</code>	$-\sum_{x \in \mathcal{X}} p(x, y) \cdot \log p(x, y)$	joint entropy of x, y
<code>\jdentyx</code>	$-\int_{\mathcal{X}, \mathcal{Y}} f(x, y) \cdot \log f(x, y) dx dy$	joint diff entropy of x, y
<code>\centyx</code>	$-\sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y x) \cdot \log p(y x)$	cond entropy y x
<code>\cdentyx</code>	$-\int_{\mathcal{X}, \mathcal{Y}} f(x, y) \cdot \log f(y x) dx dy$	cond diff entropy y x
<code>\xentpq</code>	$-\sum_{x \in \mathcal{X}} p(x) \cdot \log q(x)$	cross-entropy of p, q
<code>\kldpq</code>	$D_{KL}(p q)$	KLD between p and q
<code>\kldpqt</code>	$D_{KL}(p q_{\theta})$	KLD divergence between p and parameterized q
<code>\explogpq</code>	$\mathbb{E}_p \left[\log \frac{p(X)}{q(X)} \right]$	expected LLR of p, q (def KLD)
<code>\sumlogpq</code>	$\sum_{x \in \mathcal{X}} p(x) \cdot \log \frac{p(x)}{q(x)}$	expected LLR of p, q (def KLD)

[Back to contents](#)

ml-interpretable

Macro	Notation	Comment
<code>\pert</code>	$\tilde{\#1}^{\#2 \#3}$	command to express that for #1 the subset #2 was perturbed given subset #3
<code>\fj</code>	f_j	marginal function f_j , depending on feature j
<code>\fnj</code>	f_{-j}	marginal function f_{-j} , depending on all features but j
<code>\fS</code>	f_S	marginal function f_S depending on feature set S
<code>\fC</code>	f_C	marginal function f_C depending on feature set C
<code>\fhj</code>	\hat{f}_j	marginal function fh_j , depending on feature j
<code>\fhmj</code>	\hat{f}_{-j}	marginal function fh_{-j} , depending on all features but j
<code>\fhS</code>	\hat{f}_S	marginal function fh_S depending on feature set S
<code>\fhC</code>	\hat{f}_C	marginal function fh_C depending on feature set C
<code>\XSmat</code>	\mathbf{X}_S	Design matrix subset
<code>\XCmat</code>	\mathbf{X}_C	Design matrix subset
<code>\Xnj</code>	\mathbf{X}_{-j}	Design matrix subset $-j = \{1, \dots, j-1, j+1, \dots, p\}$
<code>\Scupj</code>	$S \cup \{j\}$	coalition S but without player j
<code>\Scupk</code>	$S \cup \{k\}$	coalition S but without player k
<code>\SsubP</code>	$S \subseteq P$	coalition S subset of P
<code>\SsubPnoj</code>	$S \subseteq P \setminus \{j\}$	coalition S subset of P without player j
<code>\SsubPnojk</code>	$S \subseteq P \setminus \{j, k\}$	coalition S subset of P without player k
<code>\phiij</code>	$\hat{\phi}_j^{(i)}$	Shapley value for feature j and observation i
<code>\Gspace</code>	\mathcal{G}	Hypothesis space for surrogate model
<code>\neigh</code>	$\phi_{\mathbf{x}}$	Proximity measure
<code>\zv</code>	\mathbf{z}	Sampled datapoints for surrogate
<code>\Zspace</code>	\mathcal{Z}	Space of sampled datapoints
<code>\Gower</code>	d_G	Gower distance

[Back to contents](#)

ml-nn

Macro	Notation	Comment
\neurons	z_1, \dots, z_M	vector of neurons
\hidz	\mathbf{z}	vector of hidden activations
\biasb	\mathbf{b}	bias vector
\biasc	c	bias in output
\wtw	\mathbf{w}	weight vector (general)
\Wmat	\mathbf{W}	weight vector (general)
\wtu	\mathbf{u}	weight vector of output neuron
\Oreg	$R_{reg}(\theta X, y)$	regularized objective function
\Ounreg	$R_{emp}(\theta X, y)$	unconstrained objective function
\Pen	$\Omega(\theta)$	penalty
\Oregweight	$R_{reg}(w X, y)$	regularized objective function with weight
\Oweight	$R_{emp}(w X, y)$	unconstrained objective function with weight
\Oweighti	$R_{emp}(w_i X, y)$	unconstrained objective function with weight w_i
\Oweightopt	$J(w^* X, y)$	unconstrained objective function with optimal weight
\Oopt	$\hat{J}(\theta X, y)$	optimal objective function
\Odropout	$J(\theta, \mu X, y)$	dropout objective function
\Loss	$L(y, f(\mathbf{x}, \boldsymbol{\theta}))$	
\Lmomentumnest	$L(y^{(i)}, f(x^{(i)}, \boldsymbol{\theta} + \varphi \boldsymbol{\nu}))$	momentum risk
\Lmomentumtilde	$L(y^{(i)}, f(x^{(i)}, \tilde{\boldsymbol{\theta}}))$	Nesterov momentum risk
\Lmomentum	$L(y^{(i)}, f(x^{(i)}, \boldsymbol{\theta}))$	
\Hess	\mathbf{H}	
\nub	$\boldsymbol{\nu}$	
\uauto	$L(x, g(f(x)))$	undercomplete autoencoder objective function
\dauto	$L(x, g(f(\tilde{x})))$	denoising autoencoder objective function
\deltab	$\boldsymbol{\delta}$	
\Lossdeltai	$L(y^{(i)}, f(\mathbf{x}^{(i)} + \boldsymbol{\delta} \boldsymbol{\theta}))$	
\Lossdelta	$L(y, f(\mathbf{x} + \boldsymbol{\delta} \boldsymbol{\theta}))$	

[Back to contents](#)

ml-survival

Macro	Notation	Comment
<code>\Ti</code>	$T^{(\#1)}$??
<code>\Ci</code>	$C^{(\#1)}$??
<code>\oi</code>	$o^{(\#1)}$??
<code>\ti</code>	$t^{(\#1)}$??
<code>\deltai</code>	$\delta^{(\#1)}$	
<code>\Lxdi</code>	$L(\boldsymbol{\delta}, f(\mathbf{x}))$	

[Back to contents](#)

ml-svm

Macro	Notation	Comment
<code>\sv</code>	SV	supportvectors
<code>\sl</code>	ζ	slack variable
<code>\slvec</code>	$(\zeta^{(1)}, \zeta^{(n)})$	slack variable vector
<code>\sli</code>	$\zeta^{(\#1)}$	i-th slack variable
<code>\scptxi</code>	$\langle \boldsymbol{\theta}, \mathbf{x}^{(i)} \rangle$	scalar prodct of theta and xi
<code>\svmhplane</code>	$y^{(i)} (\langle \boldsymbol{\theta}, \mathbf{x}^{(i)} \rangle + \theta_0)$	SVM hyperplane (normalized)
<code>\alphah</code>	$\hat{\alpha}$	alpha-hat (basis fun coefficients)
<code>\alphav</code>	$\boldsymbol{\alpha}$	vector alpha (bold) (basis fun coefficients)
<code>\alphavh</code>	$\hat{\boldsymbol{\alpha}}$	vector alpha-hat (basis fun coefficients)
<code>\dualobj</code>	$\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y^{(i)} y^{(j)} \langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle$	min objective in lin svm dual
<code>\HS</code>	Φ	H, hilbertspace
<code>\phix</code>	$\phi(\mathbf{x})$	feature map x
<code>\phixt</code>	$\phi(\tilde{\mathbf{x}})$	feature map x tilde
<code>\kxxt</code>	$k(\mathbf{x}, \tilde{\mathbf{x}})$	kernel fun x, x tilde
<code>\scptxifm</code>	$\langle \boldsymbol{\theta}, \phi(\mathbf{x}^{(i)}) \rangle$	scalar prodct of theta and xi

[Back to contents](#)

ml-trees

Macro	Notation	Comment
<code>\Np</code>	\mathcal{N}	(Parent) node N
<code>\Npk</code>	\mathcal{N}_k	node N_k
<code>\Nl</code>	\mathcal{N}_1	Left node N_1
<code>\Nr</code>	\mathcal{N}_2	Right node N_2
<code>\pikN</code>	$\pi_{\#1}^{(\mathcal{N})}$	class probability node N
<code>\pikNh</code>	$\hat{\pi}_{\#1}^{(\mathcal{N})}$	estimated class probability node N
<code>\pikNlh</code>	$\hat{\pi}_{\#1}^{(\mathcal{N}_1)}$	estimated class probability left node
<code>\pikNr</code>	$\hat{\pi}_{\#1}^{(\mathcal{N}_2)}$	estimated class probability right node

[Back to contents](#)