机器学习工具箱开发

**详细方案设计**



苏州同元软控信息技术有限公司

|  |  |  |  |
| --- | --- | --- | --- |
| 编制 |  | 生效日期 |  |
| 审核 |  | 批准 |  |

文件变更摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **日期** | **版本** | **变更说明** | **修订** | **审核** | **批准** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[1. 引言 1](#_Toc108100482)

[2. 回归 1](#_Toc108100483)

[2.1 多元线性回归 1](#_Toc108100484)

[2.1.1 多元线性回归fitlm函数 1](#_Toc108100485)

[2.1.2 多元线性回归fevel函数 2](#_Toc108100486)

[2.1.3 多元线性回归predict函数 2](#_Toc108100487)

[2.1.4 多元线性回归anova函数 3](#_Toc108100488)

[2.1.5 多元线性回归coefCI函数 4](#_Toc108100489)

[2.1.6 多元线性回归coefTest函数 5](#_Toc108100490)

[2.1.7 多元线性回归kfoldPredict函数 5](#_Toc108100491)

[2.1.8 多元线性回归dwtest函数 6](#_Toc108100492)

[2.1.9 多元线性回归partialDependence函数 7](#_Toc108100493)

[2.1.10 多元线性回归loss函数 7](#_Toc108100494)

[2.1.11 多元线性回归selectModels函数 8](#_Toc108100495)

[2.1.12 多元线性回归plsregress函数 9](#_Toc108100496)

[2.1.13 多元线性回归regress函数 10](#_Toc108100497)

[2.1.14 多元线性回归dummyvar函数 11](#_Toc108100498)

[2.1.15 多元线性回归lasso函数 11](#_Toc108100499)

[2.1.16 多元线性回归ridge函数 12](#_Toc108100500)

[2.1.17 多元线性回归fitlme函数 13](#_Toc108100501)

[2.2 广义线性回归 14](#_Toc108100502)

[2.2.1 广义线性回归fitglm函数 14](#_Toc108100503)

[2.2.2 广义线性回归predict函数 14](#_Toc108100504)

[2.2.3 广义线性回归feval函数 15](#_Toc108100505)

[2.2.4 广义线性回归coefCI函数 16](#_Toc108100506)

[2.2.5 广义线性回归coefTest函数 16](#_Toc108100507)

[2.2.6 广义线性回归fitclinear函数 17](#_Toc108100508)

[2.2.7 广义线性回归fitcecoc函数 18](#_Toc108100509)

[2.2.8 广义线性回归fitckernel函数 18](#_Toc108100510)

[2.2.9 广义线性回归mnrfit函数 19](#_Toc108100511)

[2.2.10 广义线性回归mnrval函数 20](#_Toc108100512)

[2.2.11 广义线性回归lassoglm函数 20](#_Toc108100513)

[2.3 非线性回归 21](#_Toc108100514)

[2.3.1 非线性回归fevel函数 21](#_Toc108100515)

[2.3.2 非线性回归predict函数 22](#_Toc108100516)

[2.3.3 非线性回归dummyvar函数 22](#_Toc108100517)

[2.3.4 非线性回归partialDependence函数 23](#_Toc108100518)

[2.4 SVM 24](#_Toc108100519)

[2.4.1 SVM fitrsvm函数 24](#_Toc108100520)

[2.4.2 SVM predict函数 25](#_Toc108100521)

[2.4.3 SVM crossval函数 25](#_Toc108100522)

[2.4.4 SVM partialDependence函数 26](#_Toc108100523)

[2.5 回归树 27](#_Toc108100524)

[2.5.1 回归树fitrtree函数 27](#_Toc108100525)

[2.5.2 回归树 predict函数 27](#_Toc108100526)

[2.5.3 回归树 partialDependence函数 28](#_Toc108100527)

[2.5.4 回归树predictorImportance函数 29](#_Toc108100528)

[2.5.5 回归树 crossval函数 29](#_Toc108100529)

[2.5.6 回归树kfoldPredict函数 30](#_Toc108100530)

[2.5.7 回归树kfoldLoss函数 30](#_Toc108100531)

[2.5.8 回归树loss函数 31](#_Toc108100532)

[2.5.9 回归树view函数 32](#_Toc108100533)

[2.6 高斯过程回归 32](#_Toc108100534)

[2.6.1 高斯过程回归fitrgp函数 32](#_Toc108100535)

[2.6.2 高斯过程回归predict函数 33](#_Toc108100536)

[2.6.3 高斯过程回归loss函数 34](#_Toc108100537)

[2.6.4 高斯过程回归crossval函数 34](#_Toc108100538)

[2.6.5 高斯过程回归partialDependence函数 35](#_Toc108100539)

[2.7 Lasso回归 36](#_Toc108100540)

[2.7.1 Lasso回归lassoglm函数 36](#_Toc108100541)

[2.7.2 Lasso回归predict函数 36](#_Toc108100542)

[2.8 核回归 37](#_Toc108100543)

[2.8.1 核回归fitrkernel函数 37](#_Toc108100544)

[2.8.2 核回归predict函数 38](#_Toc108100545)

[2.9 Ridge回归 38](#_Toc108100546)

[2.9.1 Ridge回归fitrkernel函数 38](#_Toc108100547)

[2.9.2 Ridge回归predict函数 39](#_Toc108100548)

[2.10 模型的构建与评估 40](#_Toc108100549)

[2.10.1 fsrftest函数 40](#_Toc108100550)

[2.10.2 oobPermutedPredictorImportance函数 40](#_Toc108100551)

[2.10.3 partialDependence函数 41](#_Toc108100552)

[2.10.4 predictorImportance函数 42](#_Toc108100553)

[2.10.5 coefCI函数 42](#_Toc108100554)

[2.10.6 coefTest函数 43](#_Toc108100555)

[2.10.7 dwtest函数 43](#_Toc108100556)

[3. 分类 44](#_Toc108100557)

[3.1 分类树 44](#_Toc108100558)

[3.1.1 分类树fitctree函数 44](#_Toc108100559)

[3.1.2 分类树partialDependence函数 45](#_Toc108100560)

[3.1.3 分类树predictorImportance函数 46](#_Toc108100561)

[3.1.4 分类树view函数 46](#_Toc108100562)

[3.1.5 分类树crossval函数 47](#_Toc108100563)

[3.1.6 分类树kfoldLoss函数 47](#_Toc108100564)

[3.1.7 分类树kfoldPredict函数 48](#_Toc108100565)

[3.1.8 分类树predict函数 49](#_Toc108100566)

[3.2 判别分析 49](#_Toc108100567)

[3.2.1 判别分析fitcdiscr函数 49](#_Toc108100568)

[3.2.2 判别分析partialDependence函数 50](#_Toc108100569)

[3.2.3 判别分析crossval函数 51](#_Toc108100570)

[3.2.4 判别分析kfoldLoss函数 51](#_Toc108100571)

[3.2.5 判别分析kfoldPredict函数 52](#_Toc108100572)

[3.2.6 判别分析predict函数 52](#_Toc108100573)

[3.3 朴素贝叶斯 53](#_Toc108100574)

[3.3.1 朴素贝叶斯fitcnb函数 53](#_Toc108100575)

[3.3.2 朴素贝叶斯partialDependence函数 54](#_Toc108100576)

[3.3.3 朴素贝叶斯crossval函数 54](#_Toc108100577)

[3.3.4 朴素贝叶斯kfoldLoss函数 55](#_Toc108100578)

[3.3.5 朴素贝叶斯kfoldPredict函数 55](#_Toc108100579)

[3.3.6 朴素贝叶斯predict函数 56](#_Toc108100580)

[3.4 最近邻 57](#_Toc108100581)

[3.4.1 最近邻fitcknn函数 57](#_Toc108100582)

[3.4.2 最近邻KDTreeSearcher函数 57](#_Toc108100583)

[3.4.3 最近邻partialDependence函数 58](#_Toc108100584)

[3.4.4 最近邻crossval函数 59](#_Toc108100585)

[3.4.5 最近邻kfoldLoss函数 59](#_Toc108100586)

[3.4.6 最近邻kfoldPredict函数 60](#_Toc108100587)

[3.4.7 最近邻predict函数 60](#_Toc108100588)

[3.4.8 最近邻pdist函数 61](#_Toc108100589)

[3.4.9 最近邻pdist2函数 62](#_Toc108100590)

[3.5 支持向量机分类 62](#_Toc108100591)

[3.5.1 支持向量机分类fitcsvm函数 62](#_Toc108100592)

[3.5.2 支持向量机分类fitSVMPosterior函数 63](#_Toc108100593)

[3.5.3 支持向量机分类predict函数 64](#_Toc108100594)

[3.5.4 支持向量机分类fitclinear函数 64](#_Toc108100595)

[3.5.5 支持向量机分类fitrkernel函数 65](#_Toc108100596)

[3.5.6 支持向量机分类fitcecoc函数 66](#_Toc108100597)

[3.6 分类集成 66](#_Toc108100598)

[3.6.1 分类集成fitcecoc函数 66](#_Toc108100599)

[3.6.2 分类集成predict函数 67](#_Toc108100600)

[3.7 用于分类的半监督学习 68](#_Toc108100601)

[3.7.1 半监督学习fitsemigraph函数 68](#_Toc108100602)

[3.7.2 半监督学习fitsemiself函数 68](#_Toc108100603)

[3.7.3 半监督学习predict函数 69](#_Toc108100604)

[3.8 模型的构建与评估 70](#_Toc108100605)

[3.8.1 fscchi2函数 70](#_Toc108100606)

[3.8.2 fscmrmr函数 70](#_Toc108100607)

[3.8.3 oobPermutedPredictorImportance函数 71](#_Toc108100608)

[3.8.4 predictorImportance函数 72](#_Toc108100609)

[3.8.5 partialDependence函数 72](#_Toc108100610)

[3.8.6 confusionchart函数 73](#_Toc108100611)

[3.8.7 confusionmat函数 74](#_Toc108100612)

[3.8.8 perfcurve函数 74](#_Toc108100613)

[4. 聚类 75](#_Toc108100614)

[4.1 层次聚类 75](#_Toc108100615)

[4.1.1 层次聚类cluster函数 75](#_Toc108100616)

[4.1.2 层次聚类clusterdata函数 76](#_Toc108100617)

[4.1.3 层次聚类linkage函数 77](#_Toc108100618)

[4.1.4 层次聚类cophenet函数 77](#_Toc108100619)

[4.1.5 层次聚类inconsistent函数 78](#_Toc108100620)

[4.1.6 层次聚类pdist函数 79](#_Toc108100621)

[4.1.7 层次聚类squareform函数 79](#_Toc108100622)

[4.2 k均值聚类和k中心点聚类 80](#_Toc108100623)

[4.2.1 k 均值聚类kmeans函数 80](#_Toc108100624)

[4.2.2 k-中心点聚类kmedoids函数 81](#_Toc108100625)

[4.2.3 马氏距离mahal函数 81](#_Toc108100626)

[4.3 基于密度的含噪数据空间聚类 82](#_Toc108100627)

[4.3.1 基于密度的含噪数据空间聚类dbscan函数 82](#_Toc108100628)

[4.4 谱聚类 83](#_Toc108100629)

[4.4.1 谱聚类spectralcluster函数 83](#_Toc108100630)

[4.5 高斯混合模型 84](#_Toc108100631)

[4.5.1 高斯混合模型cluster函数 84](#_Toc108100632)

[4.5.2 高斯混合模型gmdistribution函数 84](#_Toc108100633)

[4.5.3 高斯混合模型fitgmdist函数 85](#_Toc108100634)

[4.5.4 高斯混合模型posterior函数 86](#_Toc108100635)

[4.6 最近邻 86](#_Toc108100636)

[4.6.1 最近邻knnsearch函数 86](#_Toc108100637)

[4.6.2 最近邻KDTreeSearcher函数 87](#_Toc108100638)

[4.6.3 最近邻knnsearch函数 88](#_Toc108100639)

[4.6.4 最近邻rangesearch函数 88](#_Toc108100640)

[4.7 簇的可视化和计算 89](#_Toc108100641)

[4.7.1 dendrogram函数 89](#_Toc108100642)

[4.7.2 optimalleaforder函数 90](#_Toc108100643)

[4.8 隐马尔可夫模型 91](#_Toc108100644)

[4.8.1 隐马尔可夫模型hmmviterbi函数 91](#_Toc108100645)

[4.8.2 隐马尔可夫模型hmmestimate函数 91](#_Toc108100646)

[4.8.3 隐马尔可夫模型hmmdecode函数 92](#_Toc108100647)

[4.8.4 隐马尔可夫模型hmmtrain函数 93](#_Toc108100648)

[5. 降维和特征提取 94](#_Toc108100649)

[5.1 特征选择 94](#_Toc108100650)

[5.1.1 降维和特征提取fscchi2函数 94](#_Toc108100651)

[5.1.2 降维和特征提取fscmrmr函数 94](#_Toc108100652)

[5.1.3 降维和特征提取fsrftest函数 95](#_Toc108100653)

[5.1.4 降维和特征提取partialDependence函数 96](#_Toc108100654)

[5.1.5 降维和特征提取oobPermutedPredictorImportance函数 97](#_Toc108100655)

[5.1.6 降维和特征提取predictorImportance函数 97](#_Toc108100656)

[5.2 特征提取 98](#_Toc108100657)

[5.2.1 降维和特征提取transform函数 98](#_Toc108100658)

[5.3 PCA和典型相关 99](#_Toc108100659)

[5.3.1 降维和特征提取barttest函数 99](#_Toc108100660)

[5.3.2 降维和特征提取pca函数 99](#_Toc108100661)

[5.4 因子分析 100](#_Toc108100662)

[5.4.1 降维和特征提取factoran函数 100](#_Toc108100663)

[5.4.2 降维和特征提取rotatefactors函数 101](#_Toc108100664)

[5.5 非负矩阵分解 101](#_Toc108100665)

[5.5.1 降维和特征提取nnmf函数 101](#_Toc108100666)

[5.6 多维尺度分析 102](#_Toc108100667)

[5.6.1 降维和特征提取cmdscale函数 102](#_Toc108100668)

[5.6.2 降维和特征提取mahal函数 103](#_Toc108100669)

[5.6.3 降维和特征提取pdist函数 103](#_Toc108100670)

[5.7 普氏分析 104](#_Toc108100671)

[5.7.1 降维和特征提取procrustes函数 104](#_Toc108100672)

[6. 工业统计 105](#_Toc108100673)

[6.1 试验设计 (DOE) 105](#_Toc108100674)

[6.1.1 试验设计fullfact函数 105](#_Toc108100675)

[6.1.2 试验设计ff2n函数 106](#_Toc108100676)

[6.1.3 试验设计fracfact函数 106](#_Toc108100677)

[6.1.4 试验设计bbdesign函数 107](#_Toc108100678)

[6.1.5 试验设计ccdesign函数 108](#_Toc108100679)

[6.1.6 试验设计lhsdesign函数 108](#_Toc108100680)

[6.2 生存时间数据分析 109](#_Toc108100681)

[6.2.1 生存时间数据分析coxphfit函数 109](#_Toc108100682)

[6.2.2 生存时间数据分析ecdf函数 110](#_Toc108100683)

[6.2.3 生存时间数据分析ksdensity函数 110](#_Toc108100684)

[6.2.4 生存时间数据分析mle函数 111](#_Toc108100685)

[6.2.5 生存时间数据分析fitdist函数 112](#_Toc108100686)

[6.2.6 生存时间数据分析evfit函数 112](#_Toc108100687)

[6.2.7 生存时间数据分析expfit函数 113](#_Toc108100688)

[6.2.8 生存时间数据分析gamfit函数 113](#_Toc108100689)

[6.2.9 生存时间数据分析lognfit函数 114](#_Toc108100690)

[6.2.10 生存时间数据分析normfit函数 115](#_Toc108100691)

[6.2.11 生存时间数据分析wblfit函数 115](#_Toc108100692)

[6.3 统计过程控制 116](#_Toc108100693)

[6.3.1 统计过程控制anova1函数 116](#_Toc108100694)

[6.3.2 统计过程控制anova2函数 117](#_Toc108100695)

[6.3.3 统计过程控制anovan函数 117](#_Toc108100696)

[6.3.4 统计过程控制friedman函数 118](#_Toc108100697)

[6.3.5 统计过程控制kruskalwallis函数 119](#_Toc108100698)

[7. 绘图 119](#_Toc108100699)

[7.1 plot函数 119](#_Toc108100700)

[7.1.1 调用方式 119](#_Toc108100701)

[7.1.2 输入参数 120](#_Toc108100702)

[7.1.3 返回值 120](#_Toc108100703)

[7.1.4 开发方案 120](#_Toc108100704)

[7.2 plotPartialDependence函数 120](#_Toc108100705)

[7.2.1 调用方式 120](#_Toc108100706)

[7.2.2 输入参数 120](#_Toc108100707)

[7.2.3 返回值 120](#_Toc108100708)

[7.2.4 开发方案 121](#_Toc108100709)

[7.3 plotResiduals函数 121](#_Toc108100710)

[7.3.1 调用方式 121](#_Toc108100711)

[7.3.2 输入参数 121](#_Toc108100712)

[7.3.3 返回值 121](#_Toc108100713)

[7.3.4 开发方案 121](#_Toc108100714)

[7.4 histfit函数 121](#_Toc108100715)

[7.4.1 调用方式 121](#_Toc108100716)

[7.4.2 输入参数 122](#_Toc108100717)

[7.4.3 返回值 122](#_Toc108100718)

[7.4.4 开发方案 122](#_Toc108100719)

[8. 开发计划 122](#_Toc108100720)

# 引言

参考MATLAB统计与机器学习工具箱功能，基于julia在MWorks.Syslab平台上开发机器学习工具箱，相关核心算法调用Python或julia开源资源。

# 回归

## 多元线性回归

### 多元线性回归fitlm函数

#### 调用方式

mdl=fitlm(X, y, modelspec)

返回基于数据矩阵 X 拟合的响应 y 的线性回归模型，需提前设定线性模型modelspec的格式。

#### 输入参数

X：包含列中自变量值的矩阵。

y：包含因变量值的向量。

Modelspec：多元线性模型形式，例如@formula(y ~ x1 + x2^2)。

#### 返回值

mdl：返回模型训练结果系数，包括Coef.、Std.、Error、t Pr(>|t|)、Lower 95%、Upper 95%。

#### 开发方案

这部分主要对多元线性回归fitlm函数进行说明，本函数通过对Julia中GLM库lm()函数进行重写而来，实现了matlab多元线性回归fitlm()函数的Julia语言实现。

### 多元线性回归fevel函数

#### 调用方式

ypred = feval(mdl,Xnew1,Xnew2,...,Xnewn)

返回广义线性回归模型的响应预测ypred，每个预测值使用一个输入Xnew1,Xnew2,...,Xnewn。

#### 输入参数

mdl：fitlm()函数训练而成的模型。

Xnew1,Xnew2,...,Xnewn：包含需要预测的每个值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对多元线性回归feval函数进行说明，本函数通过对Python中sklearn库predict()函数进行重写而来，实现了matlab广义线性回归feval()函数的Julia语言实现。

### 多元线性回归predict函数

#### 调用方式

ypred = predict(mdl, Xnew)

返回需要预测的值Xnew中的点在多元线性回归模型mdl的预测响应值。

#### 输入参数

mdl：fitlm()函数训练而成的模型。

Xnew：包含需要预测的值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对多元线性回归predict函数进行说明，本函数通过对Julia中GLM库predict()函数进行重写而来，实现了matlab多元线性回归predict()函数的Julia语言实现。

### 多元线性回归anova函数

#### 调用方式

tbl = anova([mdl](https://ww2.mathworks.cn/help/stats/linearmodel.anova.html#bszh780-1_sep_shared-mdl),anovatype)

返回一个包含分量方差分析统计的表。

#### 输入参数

mdl：fitlm()或stepwiselm()函数训练而成的线性模型。

anovatype：指定为下列值之一：

①（默认）'component'—anova 返回[tbl](https://ww2.mathworks.cn/help/stats/linearmodel.anova.html#bszh780-1-tbl)包含模型中除常数项之外的每个变量的 ANOVA 统计信息的表。

②'summary'—anova 返回tbl包含分组变量和整个模型的汇总 anova 统计数据的表

#### 返回值

tbl：返回一个包含分量方差分析统计的表。

#### 开发方案

这部分主要对多元线性回归anova函数进行说明，本函数通过对python中statsmodels库anova\_lm()函数进行重写而来，实现了matlab多元线性回归anova()函数的Julia语言实现。

### 多元线性回归coefCI函数

#### 调用方式

ci = coefCI(mdl,alpha)

返回多元线性回归模型系数估计的置信区间的对应值。

#### 输入参数

mdl：fitlm()函数训练而成的模型。

alpha：最小显著性水平。

#### 返回值

ci：多元线性回归模型系数估计的置信区间的对应值，默认为95%置信区间。

#### 开发方案

这部分主要对多元线性回归coefCI函数进行说明，本函数通过对Python中statsmodels库conf\_int函数进行重写而来，实现了matlab广义线性回归coefCI ()函数的Julia语言实现。

### 多元线性回归coefTest函数

#### 调用方式

pVal = coefTest(mdl,H)

返回多元线性回归模型系数的线性假设检验的p值，默认使用F-test。

#### 输入参数

mdl：fitlm()函数训练而成的模型。

H：假设矩阵，指定为大小为 r×s 的全秩数值索引矩阵，其中 r 是要测试的系数的线性组合数，s 是系数​​的总数。。

#### 返回值

pVal：多元线性回归模型系数的线性假设检验的p值。

#### 开发方案

这部分主要对多元线性回归coefTest函数进行说明，本函数通过对Python中statsmodels库f\_test函数进行重写而来，实现了matlab广义线性回归coefTest ()函数的Julia语言实现。

### 多元线性回归kfoldPredict函数

#### 调用方式

YHat = kfoldPredict(CVMdl)

返回由K折交叉验证分类模型 CVMdl 预测的响应。

#### 输入参数

CVMdl：带有交叉验证的机器学习模型。

#### 返回值

yFit：交叉验证的预测响应，以 n × L数值数组形式返回。

#### 开发方案

这部分主要对多元线性回归kfoldPredict函数进行说明，本函数通过对Python中sklearn.model\_selection库cross\_val\_predict ()函数进行重写而来，实现了matlab 多元线性回归kfoldPredict ()函数的Julia语言实现

### 多元线性回归dwtest函数

#### 调用方式

[p](https://ww2.mathworks.cn/help/stats/linearmodel.dwtest.html#bsz4c22-1-P) = dwtest([mdl](https://ww2.mathworks.cn/help/stats/linearmodel.dwtest.html#bsz4c22-1_sep_shared-mdl))

返回关于线性回归模型残差的[Durbin-Watson 检验的](https://ww2.mathworks.cn/help/stats/linearmodel.dwtest.html#bt2mmcp-1)p值。原假设是残差

不相关，备择假设是残差是自相关的.

#### 输入参数

mdl：fitlm()或stepwiselm()函数训练而成的线性模型。

#### 返回值

p：返回关于线性回归模型残差的[Durbin-Watson 检验的](https://ww2.mathworks.cn/help/stats/linearmodel.dwtest.html#bt2mmcp-1)p值。

#### 开发方案

这部分主要对多元线性回归dwtest函数进行说明，本函数通过对python中statsmodels库durbin\_watson()函数进行重写而来，实现了matlab多元线性回归dwtest()函数的Julia语言实现。

### 多元线性回归partialDependence函数

#### 调用方式

[pd](https://ww2.mathworks.cn/help/stats/regressiontree.partialdependence.html#mw_4cf387d8-aef1-42c1-b769-2a95425354e9) = partialDependence([RegressionMdl](https://ww2.mathworks.cn/help/stats/regressiontree.partialdependence.html#mw_e5194151-ad06-445e-bf20-8cc2c5153075_sep_shared-RegressionMdl),[Vars](https://ww2.mathworks.cn/help/stats/regressiontree.partialdependence.html#mw_8994e1b1-475c-46aa-aad6-60896bd1e182))

计算RegressionMdl中列出的预测变量与使用包含预测数据Vars的回归模型预测的响应之间的部分相关性。

#### 输入参数

RegressionMdl: 训练完成的回归模型。

Vars：列出的预测变量。

#### 返回值

pd：RegressionMdl列出的预测变量与使用包含预测数据Vars的回归模型预测的响应之间的部分相关性。

#### 开发方案

这部分主要对多元线性回归partialDependence函数进行说明，本函数通过对python中sklearn库partial\_dependence()函数进行重写而来，实现了matlab多元线性回归partialDependence()函数的Julia语言实现。

### 多元线性回归loss函数

#### 调用方式

L = loss(Mdl,X,Y)

返回Mdl模型预测的损失（均方误差）。

#### 输入参数

Mdl：机器学习模型。

X: 包含列中自变量值的矩阵。

Y：包含因变量值的向量。

#### 返回值

L：返回经过训练的模型的机器学习模型的损失。

#### 开发方案

这部分主要对多元线性回归loss函数进行说明，本函数通过对Julia中GLM库lm()和predict()函数进行组合和重写而来，实现了matlab 多元线性回归loss()函数的Julia语言实现。

### 多元线性回归selectModels函数

#### 调用方式

[SubMdl = selectModels(Mdl,idx)](https://ww2.mathworks.cn/help/stats/regressionlinear.selectmodels.html#d123e859875)

从使用各种正则化强度训练的一组线性回归模型 (Mdl) 返回训练的线性回归模型的子集。索引 idx 对应于 Mdl.Lambda 中的正则化强度，并指定要返回的模型。

#### 输入参数

Mdl: 使用各种正则化强度训练的线性回归模型，指定为[RegressionLinear](https://ww2.mathworks.cn/help/stats/regressionlinear-class.html)模型对象。可以使用 fitrlinear 创建 RegressionLinear 模型对象。

虽然 Mdl 是一个模型对象，但如果 numel(Mdl.Lambda) = L ≥ 2，那么可以将 Mdl 视为 L 个训练好的模型。

idx：对应于正则化强度的索引，指定为正整数的数值向量。idx 的值必须在区间 [1,L] 内，其中 L = numel(Mdl.Lambda)。

#### 返回值

SubMdl：从使用各种正则化强度训练的一组线性回归模型 (Mdl) 返回训练的线性回归模型的子集。

#### 开发方案

这部分主要对多元线性回归selectModels函数进行说明，本函数通过对python中sklearn库SelectFromModel()函数进行重写而来，实现了matlab多元线性回归selectModels()函数的Julia语言实现。

### 多元线性回归plsregress函数

#### 调用方式

[[XL](https://ww2.mathworks.cn/help/stats/plsregress.html#mw_baca8140-fb2e-43a4-8048-e6285f5ffcf9),[YL](https://ww2.mathworks.cn/help/stats/plsregress.html#mw_51fcb6a7-9570-4abb-9c99-0fc0ae94db1b)] = plsregress([X](https://ww2.mathworks.cn/help/stats/plsregress.html#d123e729887),[Y](https://ww2.mathworks.cn/help/stats/plsregress.html#mw_745230d1-9863-47c0-ac34-b351df206d7e),[ncomp](https://ww2.mathworks.cn/help/stats/plsregress.html#mw_fbe07116-cab0-4de9-90e0-81482a353c88))

分别返回 XL 和 YL，用于矩阵 Y 在矩阵 X 中的预测变量上的偏最小二乘 (PLS) 回归。

#### 输入参数

X: 预测变量，指定为数值矩阵。X 是一个n × p矩阵，其中 n是观测值的数量， p是预测变量的数量。X 的每一行代表一个观测值，每一列代表一个变量。X 的行数必须与 Y 相同。

Y：响应变量，指定为数值矩阵。Y 是一个 n×m 矩阵，其中 n 是观察数，m 是响应变量数。Y 的每一行代表一个观测值，每一列代表一个变量。Y 中的每一行都是 X 中相应行的响应。

[ncomp](https://ww2.mathworks.cn/help/stats/plsregress.html#mw_fbe07116-cab0-4de9-90e0-81482a353c88)：分量数，指定为数值向量。默认值为min(size(X,1) – 1,size(X,2))

#### 返回值

[XL](https://ww2.mathworks.cn/help/stats/plsregress.html#mw_baca8140-fb2e-43a4-8048-e6285f5ffcf9)：预测变量载荷，以数值矩阵形式返回。

[YL](https://ww2.mathworks.cn/help/stats/plsregress.html#mw_baca8140-fb2e-43a4-8048-e6285f5ffcf9)：响应载荷，以数值矩阵形式返回。

#### 开发方案

这部分主要对多元线性回归plsregress函数进行说明，本函数通过对python中sklearn库PLSRegression()函数进行重写而来，实现了matlab多元线性回归plsregress()函数的Julia语言实现。

### 多元线性回归regress函数

#### 调用方式

[b](https://ww2.mathworks.cn/help/stats/regress.html#d123e34112) = regress([y](https://ww2.mathworks.cn/help/stats/regress.html#d123e34010),[X](https://ww2.mathworks.cn/help/stats/regress.html#d123e34043))

包含向量 y 中的响应对矩阵 X 中的预测变量的多元线性回归的系数估计值。

#### 输入参数

y: 响应数据，指定为 n×1 数值向量。y 的行对应于不同观测值。y 的行数必须与 [X](https://ww2.mathworks.cn/help/stats/regress.html#d123e34043) 的行数相同。

X：预测变量数据，指定为 n×p 数值矩阵。X 的行对应于各个观测值，列对应于预测变量。X 的行数必须与 [y](https://ww2.mathworks.cn/help/stats/regress.html#d123e34010) 的行数相同。

#### 返回值

b：多元线性回归的系数估计值。

#### 开发方案

这部分主要对多元线性回归regress函数进行说明，本函数通过对Julia中GLM库lm()函数进行重写而来，实现了matlab多元线性回归regress()函数的Julia语言实现。

### 多元线性回归dummyvar函数

#### 调用方式

[D](https://ww2.mathworks.cn/help/stats/dummyvar.html#d123e289791) = dummyvar([group](https://ww2.mathworks.cn/help/stats/dummyvar.html#d123e289719))

返回一个包含 0 和 1 的矩阵 D，其列是组中分组变量的虚拟变量。组的每一列都是一个分组变量，其值表示类别。group 的行表示所有变量的观察结果。

#### 输入参数

group: 分组变量，指定为表示单个变量内的水平的正整数向量或分类列向量、包含一个或多个分组变量的元胞数组，或表示多个变量内的水平的正整数矩阵。

#### 返回值

D：虚拟变量，以 n×s 数值矩阵形式返回，其中 n 是组的行数，s 是组的每列中的级别数之和。从左到右，D 的列是从组的第一列创建的虚拟变量，然后是从组的第二列创建的虚拟变量，依此类推。

#### 开发方案

这部分主要对多元线性回归dummyvar函数进行说明，本函数通过对python中pandas库get\_dummies()函数进行重写而来，实现了matlab多元线性回归dummyvar()函数的Julia语言实现。

### 多元线性回归lasso函数

#### 调用方式

[B](https://ww2.mathworks.cn/help/stats/lasso.html#d123e567643) = lasso([X](https://ww2.mathworks.cn/help/stats/lasso.html#d123e566485),[y](https://ww2.mathworks.cn/help/stats/lasso.html#d123e566509))

返回预测数据 X 和响应 y 的线性模型的拟合最小二乘回归系数。

#### 输入参数

X：预测变量数据，指定为数值矩阵。

y：响应数据，指定为数值向量。

#### 返回值

B：拟合系数，以数值矩阵形式返回。

#### 开发方案

这部分主要对多元线性回归lasso函数进行说明，本函数通过对Python中sklearn库Lasso ()函数进行重写而来，实现了matlab 多元线性回归lasso ()函数的Julia语言实现。

### 多元线性回归ridge函数

#### 调用方式

[B](https://ww2.mathworks.cn/help/stats/ridge.html#d123e844590) = ridge([y](https://ww2.mathworks.cn/help/stats/ridge.html#d123e844437),[X](https://ww2.mathworks.cn/help/stats/ridge.html#d123e844466),[k](https://ww2.mathworks.cn/help/stats/ridge.html#d123e844504))

返回预测数据 X 和响应 y 的线性模型的拟合最小二乘回归系数。

#### 输入参数

y：响应数据，指定为 n×1 数值向量，其中 n 是观察数。

X：预测变量数据，指定为 n×p 数值矩阵。X 的行对应于 n 个观测值，X 的列对应于 p 个预测变量。

k：岭参数，指定为数值向量。

#### 返回值

B：系数估计值，以数值矩阵形式返回。

#### 开发方案

这部分主要对多元线性回归ridge函数进行说明，本函数通过对Python中sklearn库ridgeLasso ()函数进行重写而来，实现了matlab 多元线性回归ridge ()函数的Julia语言实现。

### 多元线性回归fitlme函数

#### 调用方式

[lme](https://ww2.mathworks.cn/help/stats/fitlme.html#btyaa5y_sep_shared-lme) = fitlme([tbl](https://ww2.mathworks.cn/help/stats/fitlme.html#btyaa5y_sep_shared-tbl),[formula](https://ww2.mathworks.cn/help/stats/fitlme.html#btyaa5y-formula))

返回一个线性混合效应模型。

#### 输入参数

tbl：输入数据，包括响应变量、预测变量和分组变量，指定为表或数据集数组。

formula：模型规范的公式，指定为 'y ~ fixed + (random1|grouping1) + ... + (randomR|groupingR)' 形式的字符向量或字符串标量。

#### 返回值

lme：线性混合效果模型，以 LinearMixedModel 对象形式返回。。

#### 开发方案

这部分主要对多元线性回归fitlme函数进行说明，本函数通过对Python中statsmodels库MixedLM ()函数进行重写而来，实现了matlab 多元线性回归fitlme ()函数的Julia语言实现。

## 广义线性回归

### 广义线性回归fitglm函数

#### 调用方式

mdl=fitglm(modelspec, X, y, distr)

返回基于数据矩阵 X 和分布函数distr拟合的响应 y 的广义线性回归模型，需提前设定线性模型modelspec的格式。

#### 输入参数

Modelspec：广义线性模型形式，例如@formula(y ~ x1 + x2^2)。

X：包含列中自变量值的矩阵

y：包含因变量值的向量。

Distr：具体的分布函数，例如二项分布Binomial()

#### 返回值

mdl：返回模型训练结果系数，包括Coef.、Std.、Error、t Pr(>|t|)、Lower 95%、Upper 95%。

#### 开发方案

这部分主要对广义线性回归进行说明，本函数通过对Julia中GLM库glm()函数进行重写而来，实现了matlab广义线性回归fitglm()函数的Julia语言实现。

### 广义线性回归predict函数

#### 调用方式

ypred = predict(mdl, Xnew)

返回需要预测的值Xnew中的点在广义线性回归模型mdl的预测响应值。

#### 输入参数

mdl：fitlm()函数训练而成的模型。

Xnew：包含需要预测的值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对广义线性回归predict函数进行说明，本函数通过对Julia中GLM库predict()函数进行重写而来，实现了matlab广义线性回归predict()函数的Julia语言实现。

### 广义线性回归response函数

#### 调用方式

y = response(mdl)

返回用于拟合线性模型mdl的响应向量 y。

#### 输入参数

mdl：拟合而成的线性模型。

#### 返回值

y：返回响应向量。

#### 开发方案

这部分主要对广义线性回归response函数进行说明，本函数通过对Julia中GLM库response()函数进行重写而来，实现了matlab广义线性回归response()函数的Julia语言实现。

### 广义线性回归feval函数

#### 调用方式

ypred = feval(mdl,Xnew1,Xnew2,...,Xnewn)

返回广义线性回归模型的响应预测ypred，每个预测值使用一个输入Xnew1,Xnew2,...,Xnewn。

#### 输入参数

mdl：fitlm()函数训练而成的模型。

Xnew1,Xnew2,...,Xnewn：包含需要预测的每个值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对广义线性回归feval函数进行说明，本函数通过对Python中sklearn库predict()函数进行重写而来，实现了matlab广义线性回归feval()函数的Julia语言实现。

### 广义线性回归coefCI函数

#### 调用方式

ci = coefCI(mdl,alpha)

返回广义线性回归模型系数估计的置信区间的对应值。

#### 输入参数

mdl：fitlm()函数训练而成的模型。

alpha：最小显著性水平。

#### 返回值

ci：广义线性回归模型系数估计的置信区间的对应值，默认为95%置信区间。

#### 开发方案

这部分主要对广义线性回归coefCI函数进行说明，本函数通过对Python中statsmodels库conf\_int函数进行重写而来，实现了matlab广义线性回归coefCI ()函数的Julia语言实现。

### 广义线性回归coefTest函数

#### 调用方式

pVal = coefTest(mdl,H)

返回广义线性回归模型系数的线性假设检验的p值，默认使用F-test。

#### 输入参数

mdl：fitlm()函数训练而成的模型。

H：假设矩阵，指定为大小为 r×s 的全秩数值索引矩阵，其中 r 是要测试的系数的线性组合数，s 是系数​​的总数。。

#### 返回值

pVal：广义线性回归模型系数的线性假设检验的p值。

#### 开发方案

这部分主要对广义线性回归coefTest函数进行说明，本函数通过对Python中statsmodels库f\_test函数进行重写而来，实现了matlab广义线性回归coefTest ()函数的Julia语言实现。

### 广义线性回归fitclinear函数

#### 调用方式

mdl = fitclinear(X,y)

返回高维数据的线性分类模型拟合。

#### 输入参数

X：包含列中自变量值的矩阵

y：包含因变量值的向量。

#### 返回值

mdl：高维数据的线性分类模型拟合值。

#### 开发方案

这部分主要对广义线性回归fitclinear函数进行说明，本函数通过对Python中sklearn.linear\_model库LogisticRegressionCV()函数进行重写而来，实现了matlab广义线性回归fitclinear ()函数的Julia语言实现。

### 广义线性回归fitcecoc函数

#### 调用方式

mdl = fitcecoc (X,y)

返回基于数据矩阵 X 拟合的响应 y 的适合支持向量机或其他分类器的多类模型。

#### 输入参数

X：包含列中自变量值的矩阵

y：包含因变量值的向量。

#### 返回值

mdl：适合支持向量机或其他分类器的多类模型。

#### 开发方案

这部分主要对广义线性回归fitcecoc函数进行说明，本函数通过对Python中sklearn.svm库LinearSVC()函数进行重写而来，实现了matlab广义线性回归fitcecoc ()函数的Julia语言实现。

### 广义线性回归fitckernel函数

#### 调用方式

mdl = fitckernel(X,y)

返回基于数据矩阵 X 拟合的响应 y 的用随机特征展开拟合的高斯核分类模型。

#### 输入参数

X：包含列中自变量值的矩阵

y：包含因变量值的向量。

#### 返回值

mdl：用随机特征展开拟合的高斯核分类模型。

#### 开发方案

这部分主要对广义线性回归fitckernel函数进行说明，本函数通过对Python中sklearn.svm库SVC ()函数进行重写而来，实现了matlab广义线性回归fitckernel ()函数的Julia语言实现。

### 广义线性回归mnrfit函数

#### 调用方式

mdl = mnrfit(X,y)

返回基于数据矩阵 X 拟合的响应 y 的多项逻辑回归模型。

#### 输入参数

X：包含列中自变量值的矩阵

y：包含因变量值的向量。

#### 返回值

mdl：多项逻辑回归模型。

#### 开发方案

这部分主要对广义线性回归mnrfit函数进行说明，本函数通过对Python中sklearn.linear\_model库LogisticRegression ()函数进行重写而来，实现了matlab广义线性回归mnrfit()函数的Julia语言实现。

### 广义线性回归mnrval函数

#### 调用方式

pihat = mnrval(mdl,X,y)

返回基于数据矩阵 X 拟合的响应 y 的多项逻辑回归值。

#### 输入参数

X：包含列中自变量值的矩阵

y：包含因变量值的向量。

mdl：多项逻辑回归模型。

#### 返回值

pihat：多项逻辑回归向量值，分别为正类和负类的概率值。

#### 开发方案

这部分主要对广义线性回归mnrval函数进行说明，本函数通过对Python中sklearn.linear\_model. LogisticRegressionCV库score ()函数进行重写而来，实现了matlab广义线性回归mnrval()函数的Julia语言实现。

### 广义线性回归lassoglm函数

#### 调用方式

B = lassoglm(X,y)

返回预测变量数据 X 和响应 y 的广义线性模型的惩罚性最大似然拟合系数，其中假设 y 中的值具有正态概率分布。 B 的每一列对应于 Lambda 中的一个特定正则化系数。

#### 输入参数

X：包含列中自变量值的矩阵

y：包含因变量值的向量。

#### 返回值

B：正则化系数向量。

#### 开发方案

这部分主要对广义线性回归lassoglm函数进行说明，本函数通过对Python中sklearn.linear\_model库Lasso ()函数进行重写而来，实现了matlab广义线性回归lassoglm()函数的Julia语言实现。

## 非线性回归

### 非线性回归fevel函数

#### 调用方式

ypred = feval(mdl,Xnew1,Xnew2,...,Xnewn)

返回广义线性回归模型的响应预测ypred，每个预测值使用一个输入Xnew1,Xnew2,...,Xnewn。

#### 输入参数

mdl：fitlm()函数训练而成的模型。

Xnew1,Xnew2,...,Xnewn：包含需要预测的每个值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对非线性回归feval函数进行说明，本函数通过对Python中sklearn库predict()函数进行重写而来，实现了matlab非线性回归feval()函数的Julia语言实现。

### 非线性回归predict函数

#### 调用方式

ypred = predict(mdl, Xnew)

返回需要预测的值Xnew中的点在广义线性回归模型mdl的预测响应值。

#### 输入参数

mdl：fitlm()函数训练而成的模型。

Xnew：包含需要预测的值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对非线性回归predict函数进行说明，本函数通过对Julia中GLM库predict()函数进行重写而来，实现了matlab非线性回归predict()函数的Julia语言实现。

### 非线性回归dummyvar函数

#### 调用方式

[D](https://ww2.mathworks.cn/help/stats/dummyvar.html#d123e289791) = dummyvar([group](https://ww2.mathworks.cn/help/stats/dummyvar.html#d123e289719))

返回一个包含 0 和 1 的矩阵 D，其列是组中分组变量的虚拟变量。组的每一列都是一个分组变量，其值表示类别。group 的行表示所有变量的观察结果。

#### 输入参数

group: 分组变量，指定为表示单个变量内的水平的正整数向量或分类列向量、包含一个或多个分组变量的元胞数组，或表示多个变量内的水平的正整数矩阵。

#### 返回值

D：虚拟变量，以 n×s 数值矩阵形式返回，其中 n 是组的行数，s 是组的每列中的级别数之和。从左到右，D 的列是从组的第一列创建的虚拟变量，然后是从组的第二列创建的虚拟变量，依此类推。

#### 开发方案

这部分主要对非线性回归dummyvar函数进行说明，本函数通过对python中pandas库get\_dummies()函数进行重写而来，实现了matlab非线性回归dummyvar()函数的Julia语言实现。

### 非线性回归partialDependence函数

#### 调用方式

[pd](https://ww2.mathworks.cn/help/stats/regressiontree.partialdependence.html#mw_4cf387d8-aef1-42c1-b769-2a95425354e9) = partialDependence([RegressionMdl](https://ww2.mathworks.cn/help/stats/regressiontree.partialdependence.html#mw_e5194151-ad06-445e-bf20-8cc2c5153075_sep_shared-RegressionMdl),[Vars](https://ww2.mathworks.cn/help/stats/regressiontree.partialdependence.html#mw_8994e1b1-475c-46aa-aad6-60896bd1e182))

计算RegressionMdl中列出的预测变量与使用包含预测数据Vars的回归模型预测的响应之间的部分相关性。

#### 输入参数

RegressionMdl: 训练完成的回归模型。

Vars：列出的预测变量。

#### 返回值

pd：RegressionMdl列出的预测变量与使用包含预测数据Vars的回归模型预测的响应之间的部分相关性。

#### 开发方案

这部分主要对非线性回归partialDependence函数进行说明，本函数通过对python中sklearn库partial\_dependence()函数进行重写而来，实现了matlab非线性回归partialDependence()函数的Julia语言实现。

## SVM

### SVM fitrsvm函数

#### 调用方式

mdl= fitrsvm (X,y)

返回使用矩阵X中的预测值和向量y中的响应值训练的完整、经过训练的SVM回归模型。

#### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

mdl：返回模型训练结果。

#### 开发方案

这部分主要对SVM进行说明，本函数通过对Julia中LIBSVM库LinearSVC()函数和ScikitLearnBase库fit!()函数进行重写和组合而来，实现了matlab SVM fitrsvm函数的Julia语言实现。

### SVM predict函数

#### 调用方式

ypred = predict(mdl, Xnew)

返回需要预测的值Xnew中的点在SVM模型mdl的预测响应值。

#### 输入参数

mdl：fitrsvm()函数训练而成的模型。

Xnew：包含需要预测的值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对SVM predict函数进行说明，本函数通过对Julia中ScikitLearnBase库predict()函数进行重写而来，实现了matlab SVM predict()函数的Julia语言实现。

### SVM crossval函数

#### 调用方式

CVMdl = crossval(mdl)

从经过训练的模型 (mdl) 返回交叉验证的机器学习模型 (CVMdl)。

#### 输入参数

mdl：需要使用的机器学习模型。

#### 返回值

CVMdl：返回经过训练的模型的交叉验证的机器学习模型。

#### 开发方案

这部分主要对SVM crossval函数进行说明，本函数通过对Python中sklearn.model\_selection库cross\_val\_score()函数进行重写而来，实现了matlab SVM crossval ()函数的Julia语言实现。

### SVM partialDependence函数

#### 调用方式

pd = partialDependence(RegressionMdl ,Vars)

计算部分相关性。

#### 输入参数

RegressionMdl: 训练完成的回归模型。

Vars：列出的预测变量。

#### 返回值

pd：部分相关性值向量。

#### 开发方案

这部分主要对SVM partialDependence函数进行说明，本函数通过对Python中sklearn.inspection库partial\_dependence ()函数进行重写而来，实现了matlab SVM partialDependence()函数的Julia语言实现。

## 回归树

### 回归树fitrtree函数

#### 调用方式

mdl= fitrtree (X,y)

返回使用矩阵X中的预测值和向量y中的响应值训练的完整、经过训练的回归树模型。

#### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

mdl：返回模型训练结果列表，包括：max\_depth、min\_samples\_leaf、

min\_samples\_split、min\_purity\_increase、pruning\_purity\_threshold、n\_subfeatures、 root、Leaves、Depth。

#### 开发方案

这部分主要对回归树进行说明，本函数通过对Julia中DecisionTree库DecisionTreeRegressor()函数和ScikitLearnBase库fit!()函数进行重写和组合而来，实现了matlab回归树fitrtree函数的Julia语言实现。

### 回归树 predict函数

#### 调用方式

ypred = predict(mdl, Xnew)

返回需要预测的值Xnew中的点在回归树模型mdl的预测响应值。

#### 输入参数

mdl：fitrtree()函数训练而成的模型。

Xnew：包含需要预测的值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对回归树predict函数进行说明，本函数通过对Julia中ScikitLearnBase库predict()函数进行重写而来，实现了matlab回归树predict()函数的Julia语言实现。

### 回归树 partialDependence函数

#### 调用方式

pd = partialDependence(RegressionMdl ,Vars)

计算部分相关性。

#### 输入参数

RegressionMdl: 训练完成的回归模型。

Vars：列出的预测变量。

#### 返回值

pd：部分相关性值向量。

#### 开发方案

这部分主要对回归树 partialDependence函数进行说明，本函数通过对Python中sklearn.inspection库partial\_dependence ()函数进行重写而来，实现了matlab 回归树 partialDependence()函数的Julia语言实现。

### 回归树predictorImportance函数

#### 调用方式

imp = predictorImportance(tree)

返回回归树的预测重要性估计向量。

#### 输入参数

tree：fitrtree()函数训练而成的模型。

#### 返回值

imp：返回回归树的预测重要性估计向量。

#### 开发方案

这部分主要对回归树predictorImportance函数进行说明，本函数通过对Python中sklearn.tree.DecisionTreeRegressor库feature\_importances\_函数进行重写而来，实现了matlab回归树predictorImportance ()函数的Julia语言实现。

### 回归树 crossval函数

#### 调用方式

CVMdl = crossval(mdl)

从经过训练的模型 (mdl) 返回交叉验证的机器学习模型 (CVMdl)。

#### 输入参数

mdl：需要使用的机器学习模型。

#### 返回值

CVMdl：返回经过训练的模型的交叉验证的机器学习模型。

#### 开发方案

这部分主要对回归树crossval函数进行说明，本函数通过对Python中sklearn.model\_selection库cross\_val\_score()函数进行重写而来，实现了matlab 回归树crossval ()函数的Julia语言实现。

### 回归树kfoldPredict函数

#### 调用方式

yFit = kfoldPredict(CVMdl)

返回由K折交叉验证回归模型 CVMdl 预测的响应，默认为10折。

#### 输入参数

CVMdl：带有交叉验证的机器学习模型。

#### 返回值

yFit：返回经过训练的模型的交叉验证的机器学习模型。

#### 开发方案

这部分主要对回归树kfoldPredict函数进行说明，本函数通过对Python中sklearn.model\_selection库cross\_val\_predict ()函数进行重写而来，实现了matlab 回归树kfoldPredict ()函数的Julia语言实现。

### 回归树kfoldLoss函数

#### 调用方式

L = kfoldLoss(CVMdl)

返回由K折交叉验证回归模型 CVMdl 预测的损失（均方误差），默认为10折。

#### 输入参数

CVMdl：带有交叉验证的机器学习模型。

#### 返回值

L：返回经过训练的模型的交叉验证的机器学习模型损失。

#### 开发方案

这部分主要对回归树kfoldLoss函数进行说明，本函数通过对Python中sklearn.metrics库mean\_squared\_error()函数进行重写而来，实现了matlab 回归树kfoldLoss ()函数的Julia语言实现。

### 回归树loss函数

#### 调用方式

L = loss(Mdl,X,y)

返回Mdl模型预测的损失（均方误差）。

#### 输入参数

Mdl：机器学习模型。

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

L：返回经过训练的模型的机器学习模型的损失。

#### 开发方案

这部分主要对回归树loss函数进行说明，本函数通过对Python中sklearn.metrics库mean\_squared\_error()函数进行重写而来，实现了matlab 回归树loss()函数的Julia语言实现。

### 回归树view函数

#### 调用方式

view(tree)

返回决策树的文本描述和图结构。

#### 输入参数

tree：经过训练的决策树模型。

#### 返回值

无

#### 开发方案

这部分主要对回归树view函数进行说明，本函数通过对Python中sklearn.tree库plot\_tree函数进行重写而来，实现了matlab 回归树view()函数的Julia语言实现。

## 高斯过程回归

### 高斯过程回归fitrgp函数

#### 调用方式

mdl= fitrgp(X,y)

返回使用矩阵X中的预测值和向量y中的响应值训练的完整、经过训练的高斯过程模型。

#### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

mdl：返回模型训练结构类型。

#### 开发方案

这部分主要对高斯过程回归fitrgp函数进行说明，本函数通过对Julia中MLJScikitLearnInterface库GaussianProcessRegressor ()函数和MLJBase库fit()函数进行重写和组合而来，实现了matlab高斯过程回归fitrgp函数的Julia语言实现。

### 高斯过程回归predict函数

#### 调用方式

ypred = predict(mdl, Xnew)

返回需要预测的值Xnew中的点在高斯过程回归模型mdl的预测响应值。

#### 输入参数

mdl：fitrtree()函数训练而成的模型。

Xnew：包含需要预测的值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对高斯过程回归predict函数进行说明，本函数通过对Julia中MLJBase库predict()函数进行重写而来，实现了matlab高斯过程回归predict()函数的Julia语言实现。

### 高斯过程回归loss函数

#### 调用方式

L = loss(Mdl,X,Y)

返回Mdl模型预测的损失（均方误差）。

#### 输入参数

Mdl：机器学习模型。

X: 包含列中自变量值的矩阵。

Y：包含因变量值的向量。

#### 返回值

L：返回经过训练的模型的机器学习模型的损失。

#### 开发方案

这部分主要对高斯过程回归loss函数进行说明，本函数通过对Python中sklearn.metrics库mean\_squared\_error()函数进行重写而来，实现了matlab 高斯过程回归loss()函数的Julia语言实现。

### 高斯过程回归crossval函数

#### 调用方式

[CVMdl](https://ww2.mathworks.cn/help/stats/classificationsvm.crossval.html#mw_b591e38e-5d9a-4a25-9dba-1b5dda047b0d) = crossval([Mdl](https://ww2.mathworks.cn/help/stats/classificationsvm.crossval.html#mw_3bfcf973-44f4-43a8-a301-c8d7441ca152))

从经过训练的模型 (Mdl) 返回交叉验证（分区）机器学习模型 (CVMdl)。

#### 输入参数

Mdl：全回归模型对象 /全分类模型对象。

#### 返回值

[CVMdl](https://ww2.mathworks.cn/help/stats/classificationsvm.crossval.html#mw_b591e38e-5d9a-4a25-9dba-1b5dda047b0d)：从经过训练的模型返回交叉验证（分区）的机器学习模型。

#### 开发方案

这部分主要对高斯过程回归crossval函数进行说明，本函数通过对python中sklearn库cross\_val\_score()函数进行重写而来，实现了matlab高斯过程回归crossval()函数的Julia语言实现。

### 高斯过程回归partialDependence函数

#### 调用方式

[pd](https://ww2.mathworks.cn/help/stats/regressiontree.partialdependence.html#mw_4cf387d8-aef1-42c1-b769-2a95425354e9) = partialDependence([RegressionMdl](https://ww2.mathworks.cn/help/stats/regressiontree.partialdependence.html#mw_e5194151-ad06-445e-bf20-8cc2c5153075_sep_shared-RegressionMdl),[Vars](https://ww2.mathworks.cn/help/stats/regressiontree.partialdependence.html#mw_8994e1b1-475c-46aa-aad6-60896bd1e182))

计算RegressionMdl中列出的预测变量与使用包含预测数据Vars的回归模型预测的响应之间的部分相关性。

#### 输入参数

RegressionMdl: 训练完成的回归模型。

Vars：列出的预测变量。

#### 返回值

pd：RegressionMdl列出的预测变量与使用包含预测数据Vars的回归模型预测的响应之间的部分相关性。

#### 开发方案

这部分主要对高斯过程回归partialDependence函数进行说明，本函数通过对python中sklearn库partial\_dependence()函数进行重写而来，实现了matlab国四过程回归partialDependence()函数的Julia语言实现。

## Lasso回归

### Lasso回归lassoglm函数

#### 调用方式

mdl= lassoglm(X,y)

返回基于数据矩阵 X 拟合的响应 y 的Lasso回归模型。

#### 输入参数

X：包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

mdl：返回模型训练结果参数。

#### 开发方案

这部分主要对Lasso回归lassoglm函数进行说明，本函数通过对Python中sklearn.linear\_model库Lasso ()函数和fit()函数进行重写和组合而来，实现了matlab Lasso回归lassoglm ()函数的Julia语言实现。

### Lasso回归predict函数

#### 调用方式

ypred = predict(Xnew)

返回需要预测的值Xnew中的点在Lasso回归模型mdl的预测响应值。

#### 输入参数

Xnew：包含需要预测的值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对Lasso回归predict函数进行说明，本函数通过对Python中sklearn库predict()函数进行重写而来，实现了matlab Lasso回归predict()函数的Julia语言实现。

## 核回归

### 核回归fitrkernel函数

#### 调用方式

mdl = fitrkernel(X,y)

返回基于数据矩阵 X 拟合的响应 y 的核回归模型。

#### 输入参数

X：包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

mdl：返回模型训练结果参数。

#### 开发方案

这部分主要对核回归fitrkernel函数进行说明，本函数通过对Python中sklearn.svm库SVR()函数和fit()函数进行重写和组合而来，实现了matlab核回归fitrkernel ()函数的Julia语言实现。

### 核回归predict函数

#### 调用方式

ypred = predict(Xnew)

返回需要预测的值Xnew中的点在核回归模型mdl的预测响应值。

#### 输入参数

Xnew：包含需要预测的值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对核回归predict函数进行说明，本函数通过对Python中sklearn库predict()函数进行重写而来，实现了matlab核回归predict()函数的Julia语言实现。

## Ridge回归

### Ridge回归fitrkernel函数

#### 调用方式

mdl = ridge(X,y)

返回基于数据矩阵 X 拟合的响应 y 的Ridge回归模型。

#### 输入参数

X：包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

mdl：返回模型训练结果参数。

#### 开发方案

这部分主要对Ridge回归fitrkernel函数进行说明，本函数通过对Python中sklearn.linear\_model库Ridge ()函数进行重写而来，实现了matlab Ridge回归ridge()函数的Julia语言实现。

### Ridge回归predict函数

#### 调用方式

ypred = predict(Xnew)

返回需要预测的值Xnew中的点在Ridge回归模型mdl的预测响应值。

#### 输入参数

Xnew：包含需要预测的值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对Ridge回归predict函数进行说明，本函数通过对Python中sklearn库predict()函数进行重写而来，实现了matlab Ridge回归predict()函数的Julia语言实现。

## 模型的构建与评估

### fsrftest函数

#### 调用方式

idx = fsrftest(X,y)

返回基于F检验的回归单变量特征排序。

#### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

idx：基于F检验的回归单变量特征排序后的向量。

#### 开发方案

这部分主要对fsrftes函数进行说明，本函数通过对Python中sklearn.feature\_selection库f\_regression()函数进行重写而来，实现了matlab 模型的构建与评估fsrftest()函数的Julia语言实现。

### oobPermutedPredictorImportance函数

#### 调用方式

Imp = oobPermutedPredictorImportance(mdl)

返回回归树随机森林的预测值排列的预测值重要性估计向量。

#### 输入参数

Mdl：回归树随机森林机器学习模型。

#### 返回值

Imp：回归树随机森林的预测值排列的预测值重要性估计向量。

#### 开发方案

这部分主要对oobPermutedPredictorImportance函数进行说明，本函数通过对Python中sklearn.ensemble.RandomForestRegressor库feature\_importances\_函数进行重写而来，实现了matlab 模型的构建与评估oobPermutedPredictorImportance ()函数的Julia语言实现。

### partialDependence函数

#### 调用方式

pd = partialDependence(RegressionMdl ,Vars)

计算部分相关性。

#### 输入参数

RegressionMdl: 训练完成的回归模型。

Vars：列出的预测变量。

#### 返回值

pd：部分相关性值向量。

#### 开发方案

这部分主要对partialDependence函数进行说明，本函数通过对Python中sklearn.inspection库partial\_dependence ()函数进行重写而来，实现了matlab模型的构建与评估 partialDependence()函数的Julia语言实现。

### predictorImportance函数

#### 调用方式

imp = predictorImportance(tree)

返回回归树的预测重要性估计向量。

#### 输入参数

tree：fitrtree()函数训练而成的模型。

#### 返回值

imp：返回回归树的预测重要性估计向量。

#### 开发方案

这部分主要对predictorImportance函数进行说明，本函数通过对Python中sklearn.tree.DecisionTreeRegressor库feature\_importances\_函数进行重写而来，实现了matlab模型的构建与评估 predictorImportance ()函数的Julia语言实现。

### coefCI函数

#### 调用方式

ci = coefCI(mdl,alpha)

返回广义线性回归模型系数估计的置信区间的对应值。

#### 输入参数

mdl：fitlm()函数训练而成的模型。

alpha：最小显著性水平。

#### 返回值

ci：广义线性回归模型系数估计的置信区间的对应值，默认为95%置信区间。

#### 开发方案

这部分主要对coefCI函数进行说明，本函数通过对Python中statsmodels库conf\_int函数进行重写而来，实现了matlab模型的构建与评估 coefCI ()函数的Julia语言实现。

### coefTest函数

#### 调用方式

pVal = coefTest(mdl,H)

返回广义线性回归模型系数的线性假设检验的p值，默认使用F-test。

#### 输入参数

mdl：fitlm()函数训练而成的模型。

H：假设矩阵，指定为大小为 r×s 的全秩数值索引矩阵，其中 r 是要测试的系数的线性组合数，s 是系数​​的总数。。

#### 返回值

pVal：广义线性回归模型系数的线性假设检验的p值。

#### 开发方案

这部分主要对coefTest函数进行说明，本函数通过对Python中statsmodels库f\_test函数进行重写而来，实现了matlab 模型的构建与评估coefTest ()函数的Julia语言实现。

### dwtest函数

#### 调用方式

p = dwtest(mdl)

返回线性回归模型对象的Durbin-Watson检验值。

#### 输入参数

mdl：线性回归模型。

#### 返回值

p：线性回归模型对象的Durbin-Watson检验值。

#### 开发方案

这部分主要对dwtest函数进行说明，本函数通过对Python中statsmodels.stats.stattools库durbin\_watson函数进行重写而来，实现了matlab 模型的构建与评估dwtest()函数的Julia语言实现。

# 分类

## 分类树

### 分类树fitctree函数

#### 调用方式

mdl = fitctree(X,y)

返回使用矩阵X中的预测值和向量y中的响应值训练的完整、经过训练的分类树模型。

#### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

mdl：返回模型训练结果列表，包括：Leaves、Depth。

#### 开发方案

这部分主要对分类树fitctree函数进行说明，本函数通过对Julia中DecisionTree库build\_tree ()函数进行重写而来，实现了matlab分类树fitctree函数的Julia语言实现。

### 分类树partialDependence函数

#### 调用方式

pd = partialDependence(ClassificationMdl,Vars,Labels)

计算部分相关性。

#### 输入参数

ClassificationMdl: 训练完成的分类模型。

Vars：列出的预测变量。

#### 返回值

pd：部分相关性值向量。

#### 开发方案

这部分主要对分类树partialDependence函数进行说明，本函数通过对Python中sklearn.inspection库partial\_dependence ()函数进行重写而来，实现了matlab分类树 partialDependence()函数的Julia语言实现。

### 分类树predictorImportance函数

#### 调用方式

imp = predictorImportance(X,y)

返回分类树的预测重要性估计向量。

#### 输入参数

X：样本数据。

y：样本标签。

#### 返回值

imp：返回分类树的预测重要性估计向量。

#### 开发方案

这部分主要对分类树predictorImportance函数进行说明，本函数通过对Python中sklearn.tree.DecisionTreeClassifier库feature\_importances\_函数进行重写而来，实现了matlab分类树 predictorImportance ()函数的Julia语言实现。

### 分类树view函数

#### 调用方式

view(tree)

返回决策树的文本描述和图结构。

#### 输入参数

tree：经过训练的决策树模型。

#### 返回值

无

#### 开发方案

这部分主要对分类树view函数进行说明，本函数通过对Julia中DecisionTree库print\_tree函数进行重写而来，实现了matlab 分类树view()函数的Julia语言实现。

### 分类树crossval函数

#### 调用方式

CVMdl = crossval(mdl)

从经过训练的模型 (mdl) 返回交叉验证的机器学习模型 (CVMdl)。

#### 输入参数

mdl：需要使用的机器学习模型。

#### 返回值

CVMdl：返回经过训练的模型的交叉验证的机器学习模型。

#### 开发方案

这部分主要对分类树crossval函数进行说明，本函数通过对Python中sklearn.model\_selection库cross\_val\_score()函数进行重写而来，实现了matlab 分类树crossval ()函数的Julia语言实现。

### 分类树kfoldLoss函数

#### 调用方式

L = kfoldLoss(CVMdl)

返回由K折交叉验证分类模型 CVMdl 预测的损失，默认为10折。

#### 输入参数

CVMdl：带有交叉验证的机器学习模型。

#### 返回值

L：返回经过训练的模型的交叉验证的机器学习模型损失。

#### 开发方案

这部分主要对分类树kfoldLoss函数进行说明，本函数通过对Python中sklearn.metrics库log\_loss ()函数进行重写而来，实现了matlab 分类树kfoldLoss ()函数的Julia语言实现。

### 分类树kfoldPredict函数

#### 调用方式

yFit = kfoldPredict(CVMdl)

返回由K折交叉验证分类模型 CVMdl 预测的响应，默认为10折。

#### 输入参数

CVMdl：带有交叉验证的机器学习模型。

#### 返回值

yFit：返回经过训练的模型的交叉验证的机器学习模型。

#### 开发方案

这部分主要对分类树kfoldPredict函数进行说明，本函数通过对Python中sklearn.model\_selection库cross\_val\_predict ()函数进行重写而来，实现了matlab 分类树kfoldPredict ()函数的Julia语言实现。

### 分类树predict函数

#### 调用方式

ypred = predict(mdl, Xnew)

返回需要预测的值Xnew中的点在分类树模型mdl的预测响应值。

#### 输入参数

mdl：fitctree()函数训练而成的模型。

Xnew：包含需要预测的值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对分类树predict函数进行说明，本函数通过对Julia中DecisionTree库apply\_tree ()函数进行重写而来，实现了matlab分类树predict()函数的Julia语言实现。

## 判别分析

### 判别分析fitcdiscr函数

#### 调用方式

mdl= fitcdiscr(X,y)

返回使用矩阵X中的预测值和向量y中的响应值训练的完整、经过训练的判别分析模型。

#### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

mdl：返回模型训练结果列表。

#### 开发方案

这部分主要对判别分析fitcdiscr函数进行说明，本函数通过对Python中sklearn.discriminant\_analysis库LinearDiscriminantAnalysis ()函数和fit()函数进行重写和组合而来，实现了matlab判别分析fitcdiscr函数的Julia语言实现。

### 判别分析partialDependence函数

#### 调用方式

pd = partialDependence(RegressionMdl ,Vars)

计算部分相关性。

#### 输入参数

RegressionMdl: 训练完成的分类模型。

Vars：列出的预测变量。

#### 返回值

pd：部分相关性值向量。

#### 开发方案

这部分主要对判别分析partialDependence函数进行说明，本函数通过对Python中sklearn.inspection库partial\_dependence ()函数进行重写而来，实现了matlab判别分析partialDependence()函数的Julia语言实现。

### 判别分析crossval函数

#### 调用方式

CVMdl = crossval(mdl)

从经过训练的模型 (mdl) 返回交叉验证的机器学习模型 (CVMdl)。

#### 输入参数

mdl：需要使用的机器学习模型。

#### 返回值

CVMdl：返回经过训练的模型的交叉验证的机器学习模型。

#### 开发方案

这部分主要对判别分析crossval函数进行说明，本函数通过对Python中sklearn.model\_selection库cross\_val\_score()函数进行重写而来，实现了matlab 判别分析crossval ()函数的Julia语言实现。

### 判别分析kfoldLoss函数

#### 调用方式

L = kfoldLoss(CVMdl)

返回由K折交叉验证分类模型 CVMdl 预测的损失，默认为10折。

#### 输入参数

CVMdl：带有交叉验证的机器学习模型。

#### 返回值

L：返回经过训练的模型的交叉验证的机器学习模型损失。

#### 开发方案

这部分主要对判别分析kfoldLoss函数进行说明，本函数通过对Python中sklearn.metrics库log\_loss ()函数进行重写而来，实现了matlab 判别分析kfoldLoss ()函数的Julia语言实现。

### 判别分析kfoldPredict函数

#### 调用方式

yFit = kfoldPredict(CVMdl)

返回由K折交叉验证分类模型 CVMdl 预测的响应，默认为10折。

#### 输入参数

CVMdl：带有交叉验证的机器学习模型。

#### 返回值

yFit：返回经过训练的模型的交叉验证的机器学习模型。

#### 开发方案

这部分主要对判别分析kfoldPredict函数进行说明，本函数通过对Python中sklearn.model\_selection库cross\_val\_predict ()函数进行重写而来，实现了matlab 判别分析kfoldPredict ()函数的Julia语言实现。

### 判别分析predict函数

#### 调用方式

ypred = predict(mdl, Xnew)

返回需要预测的值Xnew中的点在判别分析模型mdl的预测响应值。

#### 输入参数

mdl：fitcdiscr ()函数训练而成的模型。

Xnew：包含需要预测的值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对判别分析predict函数进行说明，本函数通过对Python中predict ()函数进行重写而来，实现了matlab判别分析predict()函数的Julia语言实现。

## 朴素贝叶斯

### 朴素贝叶斯fitcnb函数

#### 调用方式

mdl= fitcnb(X,y)

返回使用矩阵X中的预测值和向量y中的响应值训练的完整、经过训练的朴素贝叶斯模型（默认使用高斯朴素贝叶斯）。

#### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

mdl：返回模型训练结构类型。

#### 开发方案

这部分主要对朴素贝叶斯fitcnb函数进行说明，本函数通过对Julia中NaiveBayes库GaussianNB ()函数和fit()函数进行重写和组合而来，实现了matlab朴素贝叶斯fitcnb函数的Julia语言实现。

### 朴素贝叶斯partialDependence函数

#### 调用方式

pd = partialDependence(RegressionMdl ,Vars)

计算部分相关性。

#### 输入参数

RegressionMdl: 训练完成的分类模型。

Vars：列出的预测变量。

#### 返回值

pd：部分相关性值向量。

#### 开发方案

这部分主要对朴素贝叶斯partialDependence函数进行说明，本函数通过对Python中sklearn.inspection库partial\_dependence ()函数进行重写而来，实现了matlab朴素贝叶斯partialDependence()函数的Julia语言实现。

### 朴素贝叶斯crossval函数

#### 调用方式

CVMdl = crossval(mdl)

从经过训练的模型 (mdl) 返回交叉验证的机器学习模型 (CVMdl)。

#### 输入参数

mdl：需要使用的机器学习模型。

#### 返回值

CVMdl：返回经过训练的模型的交叉验证的机器学习模型。

#### 开发方案

这部分主要对朴素贝叶斯crossval函数进行说明，本函数通过对Python中sklearn.model\_selection库cross\_val\_score()函数进行重写而来，实现了matlab 朴素贝叶斯crossval ()函数的Julia语言实现。

### 朴素贝叶斯kfoldLoss函数

#### 调用方式

L = kfoldLoss(CVMdl)

返回由K折交叉验证分类模型 CVMdl 预测的损失，默认为10折。

#### 输入参数

CVMdl：带有交叉验证的机器学习模型。

#### 返回值

L：返回经过训练的模型的交叉验证的机器学习模型损失。

#### 开发方案

这部分主要对朴素贝叶斯kfoldLoss函数进行说明，本函数通过对Python中sklearn.metrics库log\_loss ()函数进行重写而来，实现了matlab 朴素贝叶斯kfoldLoss ()函数的Julia语言实现。

### 朴素贝叶斯kfoldPredict函数

#### 调用方式

yFit = kfoldPredict(CVMdl)

返回由K折交叉验证分类模型 CVMdl 预测的响应，默认为10折。

#### 输入参数

CVMdl：带有交叉验证的机器学习模型。

#### 返回值

yFit：返回经过训练的模型的交叉验证的机器学习模型。

#### 开发方案

这部分主要对朴素贝叶斯kfoldPredict函数进行说明，本函数通过对Python中sklearn.model\_selection库cross\_val\_predict ()函数进行重写而来，实现了matlab 朴素贝叶斯kfoldPredict ()函数的Julia语言实现。

### 朴素贝叶斯predict函数

#### 调用方式

ypred = predict(mdl, Xnew)

返回需要预测的值Xnew中的点在朴素贝叶斯模型mdl的预测响应值。

#### 输入参数

mdl：fitcnb ()函数训练而成的模型。

Xnew：包含需要预测的值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对朴素贝叶斯predict函数进行说明，本函数通过对Julia中NaiveBayes库predict ()函数进行重写而来，实现了matlab朴素贝叶斯predict()函数的Julia语言实现。

## 最近邻

### 最近邻fitcknn函数

#### 调用方式

mdl= fitcknn(X,y)

返回使用矩阵X中的预测值和向量y中的响应值训练的完整、经过训练的最近邻模型。

#### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

mdl：返回模型训练结果列表，包括Number of points、Dimensions、Metric、Reordered等。

#### 开发方案

这部分主要对最近邻fitcknn函数进行说明，本函数通过对Julia中NearestNeighborModels库KNNClassifier ()函数和MLJBase库fit()函数进行重写和组合而来，实现了matlab最近邻fitcknn函数的Julia语言实现。

### 最近邻KDTreeSearcher函数

#### 调用方式

mdl = KDTreeSearcher(X)

创建Kd树最近邻搜索器。

#### 输入参数

X: 包含列中自变量值的矩阵。

#### 返回值

mdl：返回模型创建的结构类型。

#### 开发方案

这部分主要对最近邻KDTreeSearcher函数进行说明，本函数通过对Python中sklearn.neighbors库KDTree ()函数进行重写而来，实现了matlab最近邻KDTreeSearcher函数的Julia语言实现。

### 最近邻partialDependence函数

#### 调用方式

pd = partialDependence(RegressionMdl ,Vars)

计算部分相关性。

#### 输入参数

RegressionMdl: 训练完成的分类模型。

Vars：列出的预测变量。

#### 返回值

pd：部分相关性值向量。

#### 开发方案

这部分主要对最近邻partialDependence函数进行说明，本函数通过对Python中sklearn.inspection库partial\_dependence ()函数进行重写而来，实现了matlab最近邻partialDependence()函数的Julia语言实现。

### 最近邻crossval函数

#### 调用方式

CVMdl = crossval(mdl)

从经过训练的模型 (mdl) 返回交叉验证的机器学习模型 (CVMdl)。

#### 输入参数

mdl：需要使用的机器学习模型。

#### 返回值

CVMdl：返回经过训练的模型的交叉验证的机器学习模型。

#### 开发方案

这部分主要对最近邻crossval函数进行说明，本函数通过对Python中sklearn.model\_selection库cross\_val\_score()函数进行重写而来，实现了matlab 最近邻crossval ()函数的Julia语言实现。

### 最近邻kfoldLoss函数

#### 调用方式

L = kfoldLoss(CVMdl)

返回由K折交叉验证分类模型 CVMdl 预测的损失，默认为10折。

#### 输入参数

CVMdl：带有交叉验证的机器学习模型。

#### 返回值

L：返回经过训练的模型的交叉验证的机器学习模型损失。

#### 开发方案

这部分主要对最近邻kfoldLoss函数进行说明，本函数通过对Python中sklearn.metrics库log\_loss ()函数进行重写而来，实现了matlab 最近邻kfoldLoss ()函数的Julia语言实现。

### 最近邻kfoldPredict函数

#### 调用方式

yFit = kfoldPredict(CVMdl)

返回由K折交叉验证分类模型 CVMdl 预测的响应，默认为10折。

#### 输入参数

CVMdl：带有交叉验证的机器学习模型。

#### 返回值

yFit：返回经过训练的模型的交叉验证的机器学习模型。

#### 开发方案

这部分主要对最近邻kfoldPredict函数进行说明，本函数通过对Python中sklearn.model\_selection库cross\_val\_predict ()函数进行重写而来，实现了matlab 最近邻kfoldPredict ()函数的Julia语言实现。

### 最近邻predict函数

#### 调用方式

ypred = predict(mdl, Xnew)

返回需要预测的值Xnew中的点在最近邻模型mdl的预测响应值。

#### 输入参数

mdl：fitcknn ()函数训练而成的模型。

Xnew：包含需要预测的值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对最近邻predict函数进行说明，本函数通过对Julia中MLJBase库predict ()函数进行重写而来，实现了matlab最近邻predict()函数的Julia语言实现。

### 最近邻pdist函数

#### 调用方式

D = pdist(X)

返回成对观测值之间的成对距离，默认使用欧几里得空间距离。

#### 输入参数

X: 包含变量值的向量。

#### 返回值

D：返回成对观测值之间的距离值。

#### 开发方案

这部分主要对最近邻pdist函数进行说明，本函数通过对Python中scipy.spatial.distance库pdist ()函数进行重写而来，实现了matlab最近邻pdist函数的Julia语言实现。

### 最近邻pdist2函数

#### 调用方式

D = pdist2(X,Y,Distance)

使用距离指定的度量返回 X 和 Y 中每对观测值之间的距离。

#### 输入参数

X: 包含变量值的向量。

Y：包含变量值的向量。

Distance：距离度量方式：'seuclidean', 'minkowski', or 'mahalanobis'等。

#### 返回值

D：返回两组观测值之间的成对距离。

#### 开发方案

这部分主要对最近邻pdist2函数进行说明，本函数通过对Python中scipy.spatial.distance库cdist ()函数进行重写而来，实现了matlab最近邻pdist2函数的Julia语言实现。

## 支持向量机分类

### 支持向量机分类fitcsvm函数

#### 调用方式

mdl= fitcsvm (X,y)

返回使用矩阵X中的预测值和向量y中的响应值训练的完整、经过训练的支持向量机分类模型。

#### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

mdl：返回模型训练结果。

#### 开发方案

这部分主要对支持向量机分类进行说明，本函数通过对Julia中LIBSVM库svmtrain ()函数进行重写而来，实现了matlab 支持向量机分类 fitcsvm函数的Julia语言实现。

### 支持向量机分类predict函数

#### 调用方式

ypred = predict(mdl, Xnew)

返回需要预测的值Xnew中的点在支持向量机分类模型mdl的预测响应值。

#### 输入参数

mdl：fitctree()函数训练而成的模型。

Xnew：包含需要预测的值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对支持向量机分类predict函数进行说明，本函数通过对Julia中LIBSVM库svmpredict ()函数进行重写而来，实现了matlab支持向量机分类predict()函数的Julia语言实现。

### 支持向量机分类fitclinear函数

#### 调用方式

mdl = fitclinear(X,y)

返回使用矩阵X中的预测值和向量y中的响应值训练的完整、经过训练的支持向量机分类模型。

#### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

mdl：返回模型训练结构类型。

#### 开发方案

这部分主要对支持向量机分类fitclinear函数进行说明，本函数通过对Julia中sklearn.svm库linearSVC()函数和fit()函数进行重写和组合而来，实现了matlab支持向量机分类fitclinear函数的Julia语言实现。

### 支持向量机分类fitrkernel函数

#### 调用方式

mdl= fitrkernel (X,y)

返回使用矩阵X中的预测值和向量y中的响应值训练的完整、经过训练的支持向量机分类模型。

#### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

mdl：返回模型训练结构类型。

#### 开发方案

这部分主要对支持向量机分类fitrkernel函数进行说明，本函数通过对Julia中sklearn.svm库SVR()函数和fit()函数进行重写和组合而来，实现了matlab支持向量机分类fitclinear函数的Julia语言实现。

### 支持向量机分类fitcecoc函数

#### 调用方式

mdl = fitcecoc(X,y)

返回使用矩阵X中的预测值和向量y中的响应值训练的完整、经过训练的多类模型，默认为SVM。

#### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

mdl：返回模型训练结构类型。

#### 开发方案

这部分主要对支持向量机分类fitcecoc函数进行说明，本函数通过对Python中sklearn.multiclass库OutputCodeClassifier ()函数和fit()函数进行重写和组合而来，实现了matlab支持向量机分类fitcecoc函数的Julia语言实现。

## 分类集成

### 分类集成fitcecoc函数

#### 调用方式

mdl = fitcecoc(X,y)

返回使用矩阵X中的预测值和向量y中的响应值训练的完整、经过训练的多类模型，默认为SVM。

#### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

mdl：返回模型训练结构类型。

#### 开发方案

这部分主要对分类集成fitcecoc函数进行说明，本函数通过对Python中sklearn.multiclass库OutputCodeClassifier ()函数和fit()函数进行重写和组合而来，实现了matlab分类集成fitcecoc函数的Julia语言实现。

### 分类集成predict函数

#### 调用方式

ypred = predict(mdl, Xnew)

返回需要预测的值Xnew中的点多类纠错输出码（ECOC）模型mdl的预测响应值。

#### 输入参数

mdl：fitcecoc ()函数训练而成的模型。

Xnew：包含需要预测的值组成的向量。

#### 返回值

ypred：返回模型预测的结果向量。

#### 开发方案

这部分主要对分类集成predict函数进行说明，本函数通过对Julia中ScikitLearnBase库predict()函数进行重写而来，实现了matlab分类集成predict()函数的Julia语言实现。

## 用于分类的半监督学习

### 半监督学习fitsemigraph函数

#### 调用方式

mdl = fitsemigraph(X,y,UnlabeledX)

使用 X 中的预测变量数据和 y 中的标签来标记 UnlabeledX 中的数据。

#### 输入参数

X: 包含列中自变量值的矩阵（标签已知）。

y：包含因变量值的向量。

UnlabeledX：包含列中自变量值的矩阵（标签未知）。

#### 返回值

mdl：返回模型的训练结果。

#### 开发方案

这部分主要对半监督学习fitsemigraph函数进行说明，本函数通过对Python中sklearn.semi\_supervised库LabelPropagation ()函数和fit()函数进行重写和组合而来，实现了matlab半监督学习fitsemigraph函数的Julia语言实现。

### 半监督学习fitsemiself函数

#### 调用方式

mdl = fitsemiself(X,y,UnlabeledX))

使用 X 中的预测变量数据和 y 中的标签来标记 UnlabeledX 中的数据。

#### 输入参数

X: 包含列中自变量值的矩阵（标签已知）。

y：包含因变量值的向量。

UnlabeledX：包含列中自变量值的矩阵（标签未知）。

#### 返回值

mdl：返回模型的训练结果。

#### 开发方案

这部分主要对半监督学习fitsemiself函数进行说明，本函数通过对Python中sklearn.semi\_supervised库SelfTrainingClassifier ()函数和fit()函数进行重写和组合而来，实现了matlab半监督学习fitsemiself函数的Julia语言实现。

### 半监督学习predict函数

#### 调用方式

ypred = predict(mdl, Xnew)

基于半监督图的分类器标记新数据Xnew。

#### 输入参数

mdl：fitsemiself ()函数或fitsemigraph()函数训练而成的模型。

Xnew：包含需要标记的值组成的向量。

#### 返回值

ypred：返回模型标记的结果向量。

#### 开发方案

这部分主要对半监督学习predict函数进行说明，本函数通过对Julia中ScikitLearnBase库predict()函数进行重写而来，实现了matlab半监督学习predict()函数的Julia语言实现。

## 模型的构建与评估

### fscchi2函数

#### 调用方式

idx = fscchi2(X,y)

基于卡方检验的单变量特征分类排序，即使用响应变量 y对 X 中的预测变量进行排名。

#### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

idx：返回特征分类的排序结果。

#### 开发方案

这部分主要对fscchi2函数进行说明，本函数通过对Python中sklearn.feature\_selection库chi2 ()函数和sklearn.feature\_selection库SelectKBest ()函数以及fit\_transform()函数进行重写和组合而来，实现了matlab模型的构建与评估fscchi2函数的Julia语言实现。

### fscmrmr函数

#### 调用方式

idx = fscmrmr(X,y)

利用最小冗余最大相关（MRMR）算法对分类特征进行排序，即使用响应变量 y对 X 中的预测变量进行排名。

#### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

#### 返回值

idx：返回特征分类的排序结果。

#### 开发方案

这部分主要对fscmrmr函数进行说明，本函数通过对Python中pymrmr库mRMR ()函数进行重写而来，实现了matlab模型的构建与评估fscmrmr函数的Julia语言实现。

### oobPermutedPredictorImportance函数

#### 调用方式

Imp = oobPermutedPredictorImportance(mdl)

返回分类树随机森林的预测值排列的预测值重要性估计向量。

#### 输入参数

Mdl：分类树随机森林机器学习模型。

#### 返回值

Imp：分类树随机森林的预测值排列的预测值重要性估计向量。

#### 开发方案

这部分主要对oobPermutedPredictorImportance函数进行说明，本函数通过对Python中sklearn.ensemble.RandomForestClassifier库feature\_importances\_函数进行重写而来，实现了matlab 模型的构建与评估oobPermutedPredictorImportance ()函数的Julia语言实现。

### predictorImportance函数

#### 调用方式

imp = predictorImportance(tree)

返回分类树的预测重要性估计向量。

#### 输入参数

tree：fitrtree()函数训练而成的模型。

#### 返回值

imp：返回分类树的预测重要性估计向量。

#### 开发方案

这部分主要对分类树predictorImportance函数进行说明，本函数通过对Python中sklearn.tree.DecisionTreeClassifier库feature\_importances\_函数进行重写而来，实现了matlab分类树 predictorImportance ()函数的Julia语言实现。

### partialDependence函数

#### 调用方式

pd = partialDependence(RegressionMdl ,Vars)

计算部分相关性。

#### 输入参数

RegressionMdl: 训练完成的分类模型。

Vars：列出的预测变量。

#### 返回值

pd：部分相关性值向量。

#### 开发方案

这部分主要对partialDependence函数进行说明，本函数通过对Python中sklearn.inspection库partial\_dependence ()函数进行重写而来，实现了matlab模型的构建与评估partialDependence()函数的Julia语言实现。

### test函数

#### 调用方式

idx = test(c)

生成交叉验证的测试指标。

#### 输入参数

c: 验证分区，指定为 cvpartition 对象。

#### 返回值

idx：测试集观察的索引，以逻辑向量形式返回。值为 1 表示相应的观测值在测试集中。值为 0 表示对应的观测值在训练集中。

#### 开发方案

这部分主要对test函数进行说明，本函数通过对Julia中自定义函数进行编写而来，实现了matlab模型的构建与评估test()函数的Julia语言实现。

### confusionchart函数

#### 调用方式

confusionchart(trueLabels,predictedLabels)

返回真实标签trueLabels和预测标签predictedLabels所绘制的混淆矩阵图。

#### 输入参数

trueLabels: 真实标签向量。

predictedLabels：预测标签向量。

#### 返回值

无

#### 开发方案

这部分主要对confusionchart函数进行说明，本函数通过对Python中sklearn.metrics库plot\_confusion\_matrix ()函数进行重写而来，实现了matlab模型的构建与评估confusionchart函数的Julia语言实现。

### confusionmat函数

#### 调用方式

C= confusionmat (trueLabels,predictedLabels)

返回真实标签trueLabels和预测标签predictedLabels所计算的混淆矩阵。

#### 输入参数

trueLabels: 真实标签向量。

predictedLabels：预测标签向量。

#### 返回值

C：返回模型计算得到的混淆矩阵。

#### 开发方案

这部分主要对confusionmat函数进行说明，本函数通过对Python中sklearn.metrics库confusion\_matrix ()函数进行重写而来，实现了matlab模型的构建与评估confusionmat函数的Julia语言实现。

### perfcurve函数

#### 调用方式

[X,Y,T] = perfcurve(labels,scores,posclass)

返回 X 和 Y 的计算值的分类器分数阈值数组。

#### 输入参数

labels：真实的类标签。

scores：预测的类标签。

posclass：正类标签。

#### 返回值

X: ROC曲线的横坐标，即fpr。

Y：ROC曲线的纵坐标，即tpr。

T：基于fpr和tpr计算的决策函数的阈值。

#### 开发方案

这部分主要对perfcurve函数进行说明，本函数通过对Python中sklearn.metrics库roc\_curve ()函数进行重写而来，实现了matlab模型的构建与评估perfcurve函数的Julia语言实现。

1. **聚类**
   1. **层次聚类**
      1. **层次聚类cluster函数**
         1. **调用方式**

T = cluster(Z,X)

返回使用矩阵X中的层次聚类后的模型结果。

* + - 1. **输入参数**

X: 包含列中自变量值的矩阵。

Z：层次聚类参数，包括：n\_clusters、affinity、memory、connectivity、compute\_full\_tree、linkage、distance\_threshold。

* + - 1. **返回值**

T：返回模型聚类结果列表，包括：n\_clusters、labels、n\_leaves、n\_connected\_components、children。

* + - 1. **开发方案**

这部分主要对层次聚类cluster函数进行说明，本函数通过对Julia中MLJ库MLJScikitLearnInterface.AgglomerativeClustering ()函数和MLJBase库fit()函数和MLJBase库fitted\_params ()函数进行组合和重写而来，实现了matlab层次聚类cluster函数的Julia语言实现。

* + 1. **层次聚类clusterdata函数**
       1. **调用方式**

T = clusterdata(X,cutoff)

返回输入数据矩阵 X 的每个观测值（行）的聚类索引，给定用于分割链接函数从 X 生成的凝聚层次树的阈值。

* + - 1. **输入参数**

X: 包含列中自变量值的矩阵。

cutoff：分割凝聚层次树的阈值。

* + - 1. **返回值**

T：返回模型聚类结果索引列表。

* + - 1. **开发方案**

这部分主要对层次聚类clusterdata函数进行说明，本函数通过对python中scipy库cluster.hierarchy.fclusterdata ()函数进行重写而来，实现了matlab层次聚类clusterdata函数的Julia语言实现。

* + 1. **层次聚类linkage函数**
       1. **调用方式**

Z = linkage(X,method)

使用指定的方法创建树，该方法描述了如何测量集群之间的距离。

* + - 1. **输入参数**

X: 包含列中自变量值的矩阵。

method：计算集群间距离的算法，包括single、complete、average、weighted、centroid、median、ward。

* + - 1. **返回值**

Z：返回一个矩阵 Z，它对包含输入数据矩阵 X 的行的层次聚类的树进行编码。

* + - 1. **开发方案**

这部分主要对层次聚类linkage函数进行说明，本函数通过对python中scipy库cluster.hierarchy. linkage ()函数进行重写而来，实现了matlab层次聚类linkage函数的Julia语言实现。

* + 1. **层次聚类cophenet函数**
       1. **调用方式**

c = cophenet(Z)

返回计算由 Z 表示的层次聚类树的共生相关系数。

* + - 1. **输入参数**

Z:链接函数的输出。

* + - 1. **返回值**

返回计算由 Z 表示的层次聚类树的共生相关系数。

* + - 1. **开发方案**

这部分主要对层次聚类cophenet函数进行说明，本函数通过对python中scipy库cluster.hierarchy.cophenet ()函数进行重写而来，实现了matlab层次聚类cophenet函数的Julia语言实现。

* + 1. **层次聚类inconsistent函数**
       1. **调用方式**

Y = inconsistent(Z)

返回由链接函数生成的层次聚类树 Z 的每个链接的不一致系数。不一致通过将每个链接的高度与层次结构同一级别的其他链接的平均高度进行比较来计算每个链接的不一致性系数。系数越大，链接所连接的对象之间的差异越大。

* + - 1. **输入参数**

Z:链接函数的输出。

* + - 1. **返回值**

返回由链接函数生成的层次聚类树 Z 的每个链接的不一致系数。

* + - 1. **开发方案**

这部分主要对层次聚类inconsistent函数进行说明，本函数通过对python中scipy库cluster.hierarchy. inconsistent ()函数进行重写而来，实现了matlab层次聚类inconsistent函数的Julia语言实现。

* + 1. **层次聚类pdist函数**
       1. **调用方式**

D = pdist(X)

返回成对观测值之间的成对距离，默认使用欧几里得空间距离。

* + - 1. **输入参数**

X: 包含变量值的向量。

* + - 1. **返回值**

D：返回成对观测值之间的距离值。

* + - 1. **开发方案**

这部分主要对层次聚类pdist函数进行说明，本函数通过对Python中scipy.spatial.distance库pdist ()函数进行重写而来，实现了matlab层次聚类pdist函数的Julia语言实现。

* + 1. **层次聚类squareform函数**
       1. **调用方式**

ZOut = squareform(yIn)

将 yIn（用于 m 个观测的长度为 m(m–1)/2 的成对距离向量）转换为 ZOut（沿对角线具有零点的 m×m 对称矩阵）。

* + - 1. **输入参数**

yIn: 成对距离向量。

* + - 1. **返回值**

ZOut：返回沿对角线具有零点的 m×m 对称矩阵。

* + - 1. **开发方案**

这部分主要对层次聚类squareform函数进行说明，本函数通过对Python中scipy.spatial.distance库squareform ()函数进行重写而来，实现了matlab层次聚类squareform函数的Julia语言实现。

* 1. **k均值聚类和k中心点聚类**
     1. **k 均值聚类kmeans函数**
        1. **调用方式**

idx = kmeans(X,k)

将 n×p 数据矩阵 X 的观测值划分为 k 个聚类，并返回包含每个观测值的簇索引的 n×1 向量 (idx)。X 的行对应于点，列对应于变量。默认情况下，kmeans 使用欧几里德距离平方度量，并用 k-means++ 算法进行簇中心初始化。

* + - 1. **输入参数**

X: 包含列中自变量值的矩阵。

k：需要聚类的类别数。

* + - 1. **返回值**

idx：将 n×p 数据矩阵 X 的观测值划分为 k 个聚类，并返回包含每个观测值的簇索引的 n×1 向量 (idx)。

* + - 1. **开发方案**

这部分主要对k 均值聚类kmeans函数进行说明，本函数通过对Julia中Clustering库kmeans ()函数进行重写而来，实现了matlab k均值聚类kmeans函数的Julia语言实现。

* + 1. **k-中心点聚类kmedoids函数**
       1. **调用方式**

idx = kmedoids(X,k)

执行 k-medoids 聚类，将 n×p 矩阵 X 的观测值划分为 k 个簇，并返回包含每个观测值的簇索引的 n×1 向量 idx。

* + - 1. **输入参数**

X: X 的行对应于点，列对应于变量。

k：需要聚类的类别数。

* + - 1. **返回值**

idx：将 n×p 数据矩阵 X 的观测值划分为 k 个聚类，并返回包含每个观测值的簇索引的 n×1 向量 (idx)。

* + - 1. **开发方案**

这部分主要对k-中心点聚类kmedoids函数进行说明，本函数通过对Julia中Clustering库kmedoids ()函数进行重写而来，实现了matlab k-中心点聚类kmedoids函数的Julia语言实现。

* + 1. **马氏距离mahal函数**
       1. **调用方式**

d2 = mahal(Y,X)

返回 Y 中每个观测值到 X 中参考样本的平方马氏距离。

* + - 1. **输入参数**

X: 样本向量。

Y：样本向量。

* + - 1. **返回值**

d2：返回 Y 中每个观测值到 X 中参考样本的平方马氏距离。

* + - 1. **开发方案**

这部分主要对马氏距离mahal函数进行说明，本函数通过对Python中scipy.spatial.distance库mahalanobis ()函数进行重写而来，实现了matlab 马氏距离mahal函数的Julia语言实现。

* 1. **基于密度的含噪数据空间聚类**
     1. **基于密度的含噪数据空间聚类dbscan函数**
        1. **调用方式**

idx = dbscan(X,epsilon,minpts)

使用 DBSCAN 算法将 n×p 数据矩阵 X 中的观察结果划分为集群。 dbscan 基于邻域搜索半径 epsilon 的阈值和识别核心点所需的最小邻域 minpts 数对观测值（或点）进行聚类。该函数返回一个 n×1 向量 (idx)，其中包含每个观测值的集群索引。

* + - 1. **输入参数**

X: n×p 数据矩阵。

epsilon：基于邻域搜索半径 epsilon 的阈值。

minpts: 识别核心点所需的最小邻域 minpts 数对观测值。

* + - 1. **返回值**

idx：使用 DBSCAN 算法将 n×p 数据矩阵 X 中的观察结果划分为集群的结果。

* + - 1. **开发方案**

这部分主要对基于密度的含噪数据空间聚类dbscan函数进行说明，本函数通过对Julia中Clustering库dbscan()函数进行重写而来，实现了matlab 基于密度的含噪数据空间聚类dbscan函数的Julia语言实现。

* 1. **谱聚类**
     1. **谱聚类spectralcluster函数**
        1. **调用方式**

idx = spectralcluster(X,k)

使用谱聚类算法将 n×p 数据矩阵 X 中的观察结果划分为 k 个簇。 Spectrumcluster 返回一个 n×1 向量 idx ，其中包含每个观测值的集群索引。

* + - 1. **输入参数**

X: n×p 数据矩阵。

k：需要聚类的类别数。

* + - 1. **返回值**

idx：返回谱聚类的结果列表，包括affinity\_matrix、labels。

* + - 1. **开发方案**

这部分主要对谱聚类spectralcluster函数进行说明，本函数通过对Julia中MLJ库MLJScikitLearnInterface.SpectralClustering ()函数和MLJBase库fit()函数和MLJBase库fitted\_params ()函数进行组合和重写而来，实现了matlab谱聚类spectralcluster函数的Julia语言实现。

* 1. **高斯混合模型**
     1. **高斯混合模型cluster函数**
        1. **调用方式**

idx = cluster(gm,X)

将 X 中的数据划分为由 gm 中的 k 个高斯混合分量确定的 k 个簇。 idx(i) 中的值是观察 i 的聚类索引，表示在给定观察 i 的情况下具有最大后验概率的分量。

* + - 1. **输入参数**

X: n×p 数据矩阵。

gm：高斯混合模型。

* + - 1. **返回值**

idx：返回高斯混合模型聚类的结果。

* + - 1. **开发方案**

这部分主要对高斯混合模型cluster函数进行说明，本函数通过对Julia中BetaML.Clustering库gmm()函数进行重写而来，实现了matlab高斯混合模型cluster函数的Julia语言实现。

* + 1. **高斯混合模型gmdistribution函数**
       1. **调用方式**

gm = gmdistribution(mu,sigma)

使用指定的均值 mu 和具有相等混合比例的协方差 sigma 创建一个 gmdistribution 模型对象。

* + - 1. **输入参数**

mu: 指定的均值。

sigma：指定的协方差。

* + - 1. **返回值**

gm：返回gmdistribution 模型的结果列表。

* + - 1. **开发方案**

这部分主要对高斯混合模型gmdistribution函数进行说明，本函数通过对Julia中Distributions库MvNormal ()函数和MixtureModel()函数进行组合和重写而来，实现了matlab高斯混合模型gmdistribution函数的Julia语言实现。

* + 1. **高斯混合模型fitgmdist函数**
       1. **调用方式**

GMModel = fitgmdist(X,k)

返回一个高斯混合分布模型 (GMModel)，其中 k 个分量拟合到数据 (X)。

* + - 1. **输入参数**

X: n×p 数据矩阵。

k：k个分量。

* + - 1. **返回值**

GMModel：返回一个高斯混合分布模型列表，包括pₙₖ、pₖ、BIC、AIC等。

* + - 1. **开发方案**

这部分主要对高斯混合模型fitgmdist函数进行说明，本函数通过对Julia中BetaML.Clustering库gmm()函数进行重写而来，实现了matlab高斯混合模型fitgmdist函数的Julia语言实现。

* + 1. **高斯混合模型posterior函数**
       1. **调用方式**

P = posterior(gm,X)

在给定 X 中的每个观察值的情况下，返回 gm 中每个高斯混合分量的后验概率。

* + - 1. **输入参数**

X: n×p 数据矩阵。

gm：高斯混合模型。

* + - 1. **返回值**

P：返回 gm 中每个高斯混合分量的后验概率。

* + - 1. **开发方案**

这部分主要对高斯混合模型posterior函数进行说明，本函数通过对Julia中GaussianMixtures库gmmposterior ()函数进行重写而来，实现了matlab高斯混合模型posterior函数的Julia语言实现。

* 1. **最近邻**
     1. **最近邻knnsearch函数**
        1. **调用方式**

Idx = knnsearch(X,Y)

为 Y 中的每个查询点找到 X 中的最近邻，并返回 Idx（列向量）中最近邻的索引。 Idx 的行数与 Y 相同。

* + - 1. **输入参数**

X: n×p 数据矩阵。

Y: X标签的对应向量

* + - 1. **返回值**

Idx：返回最近邻训练的结果列表。

* + - 1. **开发方案**

这部分主要对最近邻knnsearch函数进行说明，本函数通过对Julia中NearestNeighbors库nn()函数和KDTree()函数进行组合和重写而来，实现了matlab最近邻knnsearch函数的Julia语言实现。

* + 1. **最近邻KDTreeSearcher函数**
       1. **调用方式**

Mdl = KDTreeSearcher(X)

使用训练数据 (X) 的 n×K 数值矩阵训练一个详尽的KD树最近邻搜索对象 (Mdl)。

* + - 1. **输入参数**

X: n×p 数据矩阵。

* + - 1. **返回值**

Mdl：返回最近邻训练的结果列表，包括Number of points、Dimensions、Metric、Reordered。

* + - 1. **开发方案**

这部分主要对最近邻KDTreeSearcher函数进行说明，本函数通过对Julia中NearestNeighbors库KDTree ()函数进行重写而来，实现了matlab最近邻KDTreeSearcher函数的Julia语言实现。

* + 1. **最近邻knnsearch函数**
       1. **调用方式**

Idx = knnsearch(Mdl,Y)

在 Mdl.X 中使用穷举搜索或 Kd-树。 knnsearch 返回 Idx，它是 Mdl.X 中表示最近邻居的索引的列向量。

* + - 1. **输入参数**

Mdl: 最近邻搜索器。

Y: X标签的对应向量

* + - 1. **返回值**

Idx：返回最近邻居的索引的列向量。

* + - 1. **开发方案**

这部分主要对最近邻knnsearch函数进行说明，本函数通过对Julia中NearestNeighbors库knn ()函数进行重写而来，实现了matlab最近邻knnsearch函数的Julia语言实现。

* + 1. **最近邻rangesearch函数**
       1. **调用方式**

Idx = rangesearch(Mdl,Y,r)

使用穷举搜索在查询数据 Y 中每个点（即行或观测值）的半径 r 内搜索 Mdl.X 中的所有邻居（即点、行或观测值）或 Kd 树。 rangesearch 返回 Idx，它是 Mdl.X 在 r 个单位内的索引的列向量。

* + - 1. **输入参数**

Mdl: 最近邻搜索器。

Y: X标签的对应向量。

r: 每个点（行或观测值）的半径。

* + - 1. **返回值**

Idx：返回最近邻居的索引的列向量。

* + - 1. **开发方案**

这部分主要对最近邻rangesearch函数进行说明，本函数通过对Julia中NearestNeighbors库inrange ()函数进行重写而来，实现了matlab最近邻rangesearch函数的Julia语言实现。

* 1. **簇的可视化和计算**
     1. **dendrogram函数**
        1. **调用方式**

dendrogram(tree)

生成分层二叉簇树的树状图。树状图由许多 U 形线组成，这些线连接层次树中的数据点。每个 U 的高度代表被连接的两个数据点之间的距离。

* + - 1. **输入参数**

tree: 分层二叉树。

* + - 1. **返回值**

无

* + - 1. **开发方案**

这部分主要对簇的可视化和计算dendrogram函数进行说明，本函数通过对Python中scipy库cluster.hierarchy.dendrogram ()函数进行重写而来，实现了matlab簇的可视化和计算dendrogram函数的Julia语言实现。

* + 1. **optimalleaforder函数**
       1. **调用方式**

leafOrder = optimalleaforder(tree,D)

使用距离 D 返回分层二叉簇树 tree 的最优叶子排序。二叉树的最优叶子排序通过在不翻转树枝的情况下最大化相邻叶子之间的相似性总和划分集群。

* + - 1. **输入参数**

tree: 分层二叉树。

D: 距离参数。

* + - 1. **返回值**

leafOrder：返回最优叶子排序的向量。

* + - 1. **开发方案**

这部分主要对optimalleaforder函数进行说明，本函数通过对Python中scipy库cluster.hierarchy.optimal\_leaf\_ordering ()函数进行重写而来，实现了matlab optimalleaforder函数的Julia语言实现。

* 1. **隐马尔可夫模型**
     1. **隐马尔可夫模型hmmviterbi函数**
        1. **调用方式**

STATES = hmmviterbi(seq,TRANS,EMIS)

给定一个序列 seq，计算通过由状态转移概率矩阵TRANS 和观测概率矩阵EMIS 指定的隐马尔可夫模型的最可能路径。 TRANS(i,j) 是从状态 i 转移到状态 j 的概率。 EMIS(i,k) 是符号 k 从状态 i 发出的概率。

* + - 1. **输入参数**

seq: 观测序列。

TRANS:状态转移概率矩阵

EMIS：观测概率矩阵

* + - 1. **返回值**

STATES: 最可能的状态序列

* + - 1. **开发方案**

这部分主要对隐马尔可夫模型hmmviterbi函数进行说明，本函数通过对Julia中HMMBase库HMM ()函数和viterbi()函数进行组合和重写而来，实现了matlab隐马尔可夫模型hmmviterbi函数的Julia语言实现。

* + 1. **隐马尔可夫模型hmmestimate函数**
       1. **调用方式**

[TRANS,EMIS] = hmmestimate(seq,TRANS,EMIS)

返回具有已知状态的序列 seq 的隐马尔可夫模型的观测概率矩阵 TRANS 和状态转移概率矩阵 EMIS 概率的最大似然估计。

* + - 1. **输入参数**

seq: 观测序列。

TRANS:状态转移概率矩阵

EMIS：观测概率矩阵

* + - 1. **返回值**

TRANS: 状态转移概率矩阵的最大似然估计值。

EMIS：观测概率矩阵的最大似然估计值。

* + - 1. **开发方案**

这部分主要对隐马尔可夫模型hmmestimate函数进行说明，本函数通过对Julia中HMMBase库HMM ()函数和loglikelihoods()函数进行组合和重写而来，实现了matlab隐马尔可夫模型hmmestimate函数的Julia语言实现。

* + 1. **隐马尔可夫模型hmmdecode函数**
       1. **调用方式**

PSTATES = hmmdecode(seq,TRANS,EMIS)

返回从隐马尔可夫模型计算序列 seq 的后验状态概率 PSTATES。

* + - 1. **输入参数**

seq: 观测序列。

TRANS:状态转移概率矩阵

EMIS：观测概率矩阵

* + - 1. **返回值**

PSTATES：后验状态概率。

* + - 1. **开发方案**

这部分主要对隐马尔可夫模型hmmdecode函数进行说明，本函数通过对Julia中HMMBase库HMM ()函数和posteriors ()函数进行组合和重写而来，实现了matlab隐马尔可夫模型hmmdecode函数的Julia语言实现。

* + 1. **隐马尔可夫模型hmmtrain函数**
       1. **调用方式**

[ESTTR,ESTEMIT] = hmmtrain(seq,TRGUESS,EMITGUESS)

使用 Baum-Welch 算法估计隐马尔可夫模型的观测概率和状态转移概率。

* + - 1. **输入参数**

seq: 观测序列。

TRGUESS: 初始观测概率矩阵。

EMITGUESS：初始状态转移概率矩阵。

* + - 1. **返回值**

ESTTR：观测概率矩阵。

ESTEMIT：状态转移概率矩阵。

* + - 1. **开发方案**

这部分主要对隐马尔可夫模型hmmtrain函数进行说明，本函数通过对Julia中HMMBase库HMM ()函数和fit\_mle ()函数进行组合和重写而来，实现了matlab隐马尔可夫模型hmmtrain函数的Julia语言实现。

### 隐马尔可夫模型hmmgenerate函数

* + - 1. **调用方式**

[seq,states] = hmmgenerate(len,TRANS,EMIS)

采用已知的马尔可夫模型，由转移概率矩阵 TRANS 和观测概率矩阵 EMIS 指定，并使用它生成输出观测的随机序列 seq；随机序列状态states。

* + - 1. **输入参数**

len: 序列长度。

TRANS:状态转移概率矩阵。

EMIS：观测概率矩阵。

* + - 1. **返回值**

seq：输出观测的随机序列。

states：随机序列状态。

* + - 1. **开发方案**

这部分主要对隐马尔可夫模型hmmgenerate函数进行说明，本函数通过对Julia中HMMBase库HMM ()函数和rand ()函数进行组合和重写而来，实现了matlab隐马尔可夫模型hmmgenerate函数的Julia语言实现。

# 降维和特征提取

## 特征选择

### 降维和特征提取fscchi2函数

#### 调用方式

[idx = fscchi2(X,Y)](https://ww2.mathworks.cn/help/stats/fscchi2.html#d123e432719)

使用响应变量 Y 对 X 中的预测变量进行排名。

#### 输入参数

X：预测变量数据，指定为数值矩阵。

y：响应变量。

#### 返回值

idx：使用响应变量 Y 对 X 中的预测变量进行排名。

#### 开发方案

这部分主要对降维和特征提取fscchi2函数进行说明，本函数通过对Python中sklearn库chi2()函数进行重写而来，实现了matlab降维和特征提取fscchi2()函数的Julia语言实现。

### 降维和特征提取fscmrmr函数

#### 调用方式

[idx](https://ww2.mathworks.cn/help/stats/fscmrmr.html#mw_c8828fd8-dea1-4a76-b893-d0cdbc8a05fc_sep_shared-idx) = fscmrmr([Tbl](https://ww2.mathworks.cn/help/stats/fscmrmr.html#mw_c8828fd8-dea1-4a76-b893-d0cdbc8a05fc_sep_shared-Tbl),[ResponseVarName](https://ww2.mathworks.cn/help/stats/fscmrmr.html#mw_c8828fd8-dea1-4a76-b893-d0cdbc8a05fc_sep_shared-ResponseVarName))

使用 MRMR 算法对特征（预测变量）进行排名。

#### 输入参数

[Tbl](https://ww2.mathworks.cn/help/stats/fscmrmr.html#mw_c8828fd8-dea1-4a76-b893-d0cdbc8a05fc_sep_shared-Tbl)：样本数据，指定为表格。

[ResponseVarName](https://ww2.mathworks.cn/help/stats/fscmrmr.html#mw_c8828fd8-dea1-4a76-b893-d0cdbc8a05fc_sep_shared-ResponseVarName)：响应变量名称，指定为包含 Tbl 中变量名称的字符向量或字符串标量。

#### 返回值

idx：使用 MRMR 算法对特征（预测变量）进行排名。

#### 开发方案

这部分主要对降维和特征提取fscmrmr函数进行说明，本函数通过对Python中pymrmr库mRMR()函数进行重写而来，实现了matlab降维和特征提取fscmrmr()函数的Julia语言实现。

### 降维和特征提取fsrftest函数

#### 调用方式

[idx](https://ww2.mathworks.cn/help/stats/fsrftest.html#mw_93dd4853-5202-41a5-af72-815f54b00ee6_sep_shared-idx) = fsrftest([Tbl](https://ww2.mathworks.cn/help/stats/fsrftest.html#mw_93dd4853-5202-41a5-af72-815f54b00ee6_sep_mw_98c996ec-017b-4d55-825d-ef28884cac63),[ResponseVarName](https://ww2.mathworks.cn/help/stats/fsrftest.html#mw_93dd4853-5202-41a5-af72-815f54b00ee6_sep_shared-ResponseVarName))

使用 F 检验对特征（预测变量）进行排名。

#### 输入参数

[Tbl](https://ww2.mathworks.cn/help/stats/fscmrmr.html#mw_c8828fd8-dea1-4a76-b893-d0cdbc8a05fc_sep_shared-Tbl)：样本数据，指定为表格。

[ResponseVarName](https://ww2.mathworks.cn/help/stats/fscmrmr.html#mw_c8828fd8-dea1-4a76-b893-d0cdbc8a05fc_sep_shared-ResponseVarName)：响应变量名称，指定为包含 Tbl 中变量名称的字符向量或字符串标量。

#### 返回值

idx：使用 F 检验对特征（预测变量）进行排名。

#### 开发方案

这部分主要对降维和特征提取fsrftest函数进行说明，本函数通过对Python中sklearn库f\_regression()函数进行重写而来，实现了matlab降维和特征提取fsrftest()函数的Julia语言实现。

### 降维和特征提取partialDependence函数

#### 调用方式

[pd](https://ww2.mathworks.cn/help/stats/regressiontree.partialdependence.html#mw_4cf387d8-aef1-42c1-b769-2a95425354e9) = partialDependence([RegressionMdl](https://ww2.mathworks.cn/help/stats/regressiontree.partialdependence.html#mw_e5194151-ad06-445e-bf20-8cc2c5153075_sep_shared-RegressionMdl),[Vars](https://ww2.mathworks.cn/help/stats/regressiontree.partialdependence.html#mw_8994e1b1-475c-46aa-aad6-60896bd1e182))

计算 Vars 中列出的预测变量与使用包含预测数据的回归模型 RegressionMdl 预测的响应之间的部分相关性。

#### 输入参数

[RegressionMdl](https://ww2.mathworks.cn/help/stats/regressiontree.partialdependence.html#mw_e5194151-ad06-445e-bf20-8cc2c5153075_sep_shared-RegressionMdl)：回归模型。

[Vars](https://ww2.mathworks.cn/help/stats/regressiontree.partialdependence.html#mw_8994e1b1-475c-46aa-aad6-60896bd1e182)：预测变量。

#### 返回值

pd：计算 Vars 中列出的预测变量与使用包含预测数据的回归模型 RegressionMdl 预测的响应之间的部分相关性。

#### 开发方案

这部分主要对降维和特征提取partialDependence函数进行说明，本函数通过对Python中sklearn库partial\_dependence()函数进行重写而来，实现了matlab降维和特征提取partialDependence()函数的Julia语言实现。

### 降维和特征提取oobPermutedPredictorImportance函数

#### 调用方式

[Imp](https://ww2.mathworks.cn/help/stats/classificationbaggedensemble.oobpermutedpredictorimportance.html#bvf730a-Imp) = oobPermutedPredictorImportance([Mdl](https://ww2.mathworks.cn/help/stats/classificationbaggedensemble.oobpermutedpredictorimportance.html#bvf730a-Mdl))

使用分类树的随机森林 Mdl 通过排列返回袋外预测器重要性估计的向量。

#### 输入参数

[Mdl](https://ww2.mathworks.cn/help/stats/regressiontree.partialdependence.html#mw_e5194151-ad06-445e-bf20-8cc2c5153075_sep_shared-RegressionMdl)：分类树的随机森林。

#### 返回值

Imp：计算 Vars 中列出的预测变量与使用包含预测数据的回归模型 RegressionMdl 预测的响应之间的部分相关性。

#### 开发方案

这部分主要对降维和特征提取oobPermutedPredictorImportance函数进行说明，本函数通过对Python中sklearn库feature\_importances\_()函数进行重写而来，实现了matlab降维和特征提取oobPermutedPredictorImportance()函数的Julia语言实现。

### 降维和特征提取predictorImportance函数

#### 调用方式

imp = predictorImportance(tree)

使用分类树的随机森林 Mdl 通过排列返回袋外预测器重要性估计的向量。

#### 输入参数

tree：由 fitctree 或 compact 方法创建的分类树。

#### 返回值

Imp：具有与树中预测变量（列）数相同的元素数的行向量。

#### 开发方案

这部分主要对降维和特征提取predictorImportance函数进行说明，本函数通过对Python中sklearn库feature\_importances\_()函数进行重写而来，实现了matlab降维和特征提取predictorImportance()函数的Julia语言实现。

## 特征提取

### 降维和特征提取transform函数

#### 调用方式

[z](https://ww2.mathworks.cn/help/stats/reconstructionica.transform.html#bvlqru_-z) = transform([Mdl](https://ww2.mathworks.cn/help/stats/reconstructionica.transform.html#bvlqru_-Mdl),[x](https://ww2.mathworks.cn/help/stats/reconstructionica.transform.html#bvlqru_-x))

通过模型 Mdl 将数据 x 转换为特征 z。

#### 输入参数

Mdl：特征提取模型。

[x](https://ww2.mathworks.cn/help/stats/reconstructionica.transform.html#bvlqru_-x)：预测变量数据，指定为具有 p 列的矩阵或具有 p 列的数值表。

#### 返回值

Imp：具有与树中预测变量（列）数相同的元素数的行向量。

#### 开发方案

这部分主要对降维和特征提取predictorImportance函数进行说明，本函数通过对Python中sklearn库feature\_importances\_()函数进行重写而来，实现了matlab降维和特征提取predictorImportance()函数的Julia语言实现。

## PCA和典型相关

### 降维和特征提取barttest函数

#### 调用方式

[ndim](https://ww2.mathworks.cn/help/stats/barttest.html#bt5e95n-ndim) = barttest([x](https://ww2.mathworks.cn/help/stats/barttest.html#bt5e95n-x),[alpha](https://ww2.mathworks.cn/help/stats/barttest.html#bt5e95n-alpha))

返回解释数据矩阵 x 在 alpha 显着性水平上的非随机变化所需的维数。

#### 输入参数

[x](https://ww2.mathworks.cn/help/stats/reconstructionica.transform.html#bvlqru_-x)：输入数据，指定为标量值矩阵。

[alpha](https://ww2.mathworks.cn/help/stats/barttest.html#bt5e95n-alpha)：假设检验的显着性水平。

#### 返回值

[ndim](https://ww2.mathworks.cn/help/stats/barttest.html#bt5e95n-ndim)：返回解释数据矩阵 x 在 alpha 显着性水平上的非随机变化所需的维数。

#### 开发方案

这部分主要对降维和特征提取barttest函数进行说明，本函数通过对Python中scipy库bartlett()函数进行重写而来，实现了matlab降维和特征提取barttest()函数的Julia语言实现。

### 降维和特征提取pca函数

#### 调用方式

[coeff](https://ww2.mathworks.cn/help/stats/pca.html#bth9ibe-coeff) = pca([X](https://ww2.mathworks.cn/help/stats/pca.html#bth9ibe-X))

返回 n×p 数据矩阵 X 的主成分系数，也称为载荷。

#### 输入参数

[X](https://ww2.mathworks.cn/help/stats/pca.html#bth9ibe-X)：计算主成分时所基于的输入数据，指定为 n×p 矩阵。

#### 返回值

[coeff](https://ww2.mathworks.cn/help/stats/pca.html#bth9ibe-coeff)：主成分系数，以 p×p 矩阵形式返回。

#### 开发方案

这部分主要对降维和特征提取pca函数进行说明，本函数通过对julia中PCA()函数进行重写而来，实现了matlab降维和特征提取pca()函数的Julia语言实现。

## 因子分析

### 降维和特征提取factoran函数

#### 调用方式

[lambda](https://ww2.mathworks.cn/help/stats/factoran.html#mw_f952ae5d-f544-4300-9922-f88f1b240456) = factoran([X](https://ww2.mathworks.cn/help/stats/factoran.html#mw_4fbfc9e6-315f-454b-961b-c028d02c3202),[m](https://ww2.mathworks.cn/help/stats/factoran.html#mw_8f8be61c-be6a-43d5-b59e-6bf1f4c11225))

返回具有 m 个公因子的数据矩阵 X 的因子载荷矩阵。

#### 输入参数

[X](https://ww2.mathworks.cn/help/stats/factoran.html#mw_4fbfc9e6-315f-454b-961b-c028d02c3202)：数据。

m：公因子数，指定为正整数。

#### 返回值

[lambda](https://ww2.mathworks.cn/help/stats/factoran.html#mw_f952ae5d-f544-4300-9922-f88f1b240456) ：返回具有 m 个公因子的数据矩阵 X 的因子载荷矩阵。

#### 开发方案

这部分主要对降维和特征提取factoran函数进行说明，本函数通过对Julia中FactorAnalysis()函数进行重写而来，实现了matlab降维和特征提取factoran()函数的Julia语言实现。

### 降维和特征提取rotatefactors函数

#### 调用方式

B = rotatefactors(A)

旋转 d×m 载荷矩阵 A 以最大化 varimax 准则，并在 B 中返回结果。

#### 输入参数

A：载荷矩阵。

#### 返回值

B：旋转 d×m 载荷矩阵 A 以最大化 varimax 准则，并在 B 中返回结果。

#### 开发方案

这部分主要对降维和特征提取rotatefactors函数进行说明，本函数通过对Python中factor\_analyzer.rotator.Rotator函数进行重写而来，实现了matlab降维和特征提取rotatefactors()函数的Julia语言实现。

## 非负矩阵分解

### 降维和特征提取nnmf函数

#### 调用方式

[[W](https://ww2.mathworks.cn/help/stats/nnmf.html#mw_ec0f8fff-e737-4ce4-8e4e-75c37d524f8f),[H](https://ww2.mathworks.cn/help/stats/nnmf.html#mw_c9d5194b-fc51-4ed8-9e1b-8d032a8661f8)] = nnmf([A](https://ww2.mathworks.cn/help/stats/nnmf.html#mw_4ca13dfb-389a-45b3-b097-bffa5d0272b4),[k](https://ww2.mathworks.cn/help/stats/nnmf.html#mw_3ebb9512-9883-433e-932e-7436dbc74361))

将 n×m 矩阵 A 分解为非负因子 W (n×k) 和 H (k×m)。

#### 输入参数

A：要分解的矩阵，指定为实矩阵。

k：因子的等级，指定为正整数。

#### 返回值

W ：A 的非负左因子，以 n×k 矩阵形式返回。

H ：A 的非负右因子，以 k×m 矩阵形式返回。

#### 开发方案

这部分主要对降维和特征提取nnmf函数进行说明，本函数通过对Python中sklearn库NMF()函数进行重写而来，实现了matlab降维和特征提取nnmf()函数的Julia语言实现。

## 多维尺度分析

### 降维和特征提取cmdscale函数

#### 调用方式

Y = cmdscale(D)

返回一个 n×p 配置矩阵 Y。

#### 输入参数

D： n×n 矩阵。

#### 返回值

Y ：返回一个 n×p 配置矩阵 Y。

#### 开发方案

这部分主要对降维和特征提取cmdscale函数进行说明，本函数通过对Python中sklearn库MDS()函数进行重写而来，实现了matlab降维和特征提取cmdscale()函数的Julia语言实现

### 降维和特征提取mahal函数

#### 调用方式

[d2](https://ww2.mathworks.cn/help/stats/mahal.html#d123e621297) = mahal([Y](https://ww2.mathworks.cn/help/stats/mahal.html#d123e621205),[X](https://ww2.mathworks.cn/help/stats/mahal.html#d123e621248))

返回 Y 中每个观测值到 X 中参考样本的平方马氏距离。

#### 输入参数

Y：数据，指定为 n×m 数值矩阵，其中 n 是观察数，m 是每个观察中的变量数。

X：参考样本，指定为 p×m 数值矩阵，其中 p 是样本数，m 是每个样本中的变量数。

#### 返回值

d2：返回 Y 中每个观测值到 X 中参考样本的平方马氏距离。

#### 开发方案

这部分主要对降维和特征提取mahal函数进行说明，本函数通过对Python中scipy库cdist()函数进行重写而来，实现了matlab降维和特征提取mahal()函数的Julia语言实现

### 降维和特征提取pdist函数

#### 调用方式

[D](https://ww2.mathworks.cn/help/stats/pdist.html#d123e30680) = pdist([X](https://ww2.mathworks.cn/help/stats/pdist.html#d123e30376))

返回 X 中成对观测值之间的欧几里德距离。

#### 输入参数

X：输入数据，指定为大小是 m×n 的数值矩阵。行对应于单个观测值，列对应单个变量。

#### 返回值

D：返回 X 中成对观测值之间的欧几里德距离。

#### 开发方案

这部分主要对降维和特征提取pdist函数进行说明，本函数通过对Python中math库dist ()函数进行重写而来，实现了matlab降维和特征提取pdist()函数的Julia语言实现

## 普氏分析

### 降维和特征提取procrustes函数

#### 调用方式

[d](https://ww2.mathworks.cn/help/stats/procrustes.html#mw_27fcfff4-ff79-4f9b-9190-7d9be790f790) = procrustes([X](https://ww2.mathworks.cn/help/stats/procrustes.html#mw_4d17683e-6827-4c36-9088-50f379b5fbe5),[Y](https://ww2.mathworks.cn/help/stats/procrustes.html#mw_21f3136c-c643-429a-b2b9-49fc16613816))

返回 X 和 Y 的形状之间的 Procrustes 距离。

#### 输入参数

X：目标形状，指定为 n×p 矩阵，其中 n 行中的每一行都包含一个 p 维界标点。

Y：比较形状，指定为 n×q 矩阵，其中 n 行中的每一行都包含 q ≤ p 的 q 维界标点。

#### 返回值

d：返回 X 和 Y 的形状之间的 Procrustes 距离。

#### 开发方案

这部分主要对降维和特征提取procrustes函数进行说明，本函数通过对Python中scipy库procrustes ()函数进行重写而来，实现了matlab降维和特征提取procrustes()函数的Julia语言实现

1. **工业统计**
   1. **试验设计 (DOE)**
      1. **试验设计fullfact函数**
         1. **调用方式**

dFF = fullfact(levels)

给出具有 n 个因子的完全析因设计的因子设置 dFF，其中每个因子的水平数由长度为 n 的向量 levels 给出。

* + - 1. **输入参数**

levels: 因子水平数。

* + - 1. **返回值**

dFF：dFF 是 m×n，其中 m 是完全析因设计中的处理数。dFF 的每行对应一种处理。每列都包含单个因子的设置，即从 1 到水平数的整数值。

* + - 1. **开发方案**

这部分主要对试验设计fullfact函数进行说明，本函数通过对Python中pyDOE库fullfact ()函数进行重写而来，实现了matlab 试验设计fullfact函数的Julia语言实现。

* + 1. **试验设计ff2n函数**
       1. **调用方式**

dFF2 = ff2n(n)

给出具有 n 个因子的二水平完全析因设计的因子设置 dFF2。

* + - 1. **输入参数**

n: 因子数。

* + - 1. **返回值**

dFF2：dFF2 为 m×n，其中 m 是完全析因设计中的处理数。dFF2 的每行对应一种处理。每列包含单一因子的设置，两个水平的值分别为 0 和 1。

* + - 1. **开发方案**

这部分主要对试验设计ff2n函数进行说明，本函数通过对Python中pyDOE库ff2n ()函数进行重写而来，实现了matlab 试验设计ff2n函数的Julia语言实现。

* + 1. **试验设计fracfact函数**
       1. **调用方式**

X = fracfact(gen)

创建由生成器 gen 定义的两水平部分因子设计。

* + - 1. **输入参数**

gen: 字符串数组或字符向量元胞数组，其中每个元素包含一个“单词”，或者由空格分隔的“单词”组成的字符数组或字符串标量。 “单词”由区分大小写的字母或字母组组成，其中“a”表示值 1，“b”表示值 2，...，“A”表示值 27，...，“Z”表示值 52 .每个词都定义了如何将相应因子的水平定义为 2^K 全因子设计的生成器的乘积。 K 是 gen 中字母的个数。

* + - 1. **返回值**

X：两水平部分因子设计，X 是一个大小为 N×P 的矩阵。

* + - 1. **开发方案**

这部分主要对试验设计fracfact函数进行说明，本函数通过对Python中pyDOE库fracfact ()函数进行重写而来，实现了matlab 试验设计fracfact函数的Julia语言实现。

* + 1. **试验设计bbdesign函数**
       1. **调用方式**

dBB = bbdesign(n)

为 n 个因子生成 Box-Behnken 设计。 n 必须是 3 或更大的整数。输出矩阵 dBB 为 m×n，其中 m 是设计中的游程数。每行代表一次运行，列中表示所有因子的设置。

* + - 1. **输入参数**

n: 因子数。

* + - 1. **返回值**

dBB：输出矩阵 dBB 为 m×n，其中 m 是设计中的游程数。每行代表一次运行，列中表示所有因子的设置，因子值已标准化，因此矩阵各点值介于 -1 和 1 之间。

* + - 1. **开发方案**

这部分主要对试验设计bbdesign函数进行说明，本函数通过对Python中pyDOE库bbdesign ()函数进行重写而来，实现了matlab 试验设计bbdesign函数的Julia语言实现。

* + 1. **试验设计ccdesign函数**
       1. **调用方式**

dCC = ccdesign(n)

为 n 个因子生成中心复合设计。 n 必须是 2 或更大的整数。输出矩阵 dCC 是 m×n，其中 m 是设计中的游程数。每行代表一次运行，列中表示所有因子的设置。因子值已标准化，因此矩阵各点的值介于 -1 和 1 之间。

* + - 1. **输入参数**

n: 因子数。

* + - 1. **返回值**

dCC：输出矩阵 dCC 是 m×n，其中 m 是设计中的游程数。每行代表一次运行，列中表示所有因子的设置。因子值已标准化，因此矩阵各点的值介于 -1 和 1 之间。

* + - 1. **开发方案**

这部分主要对试验设计ccdesign函数进行说明，本函数通过对Python中pyDOE库ccdesign ()函数进行重写而来，实现了matlab 试验设计ccdesign函数的Julia语言实现。

* + 1. **试验设计lhsdesign函数**
       1. **调用方式**

X = lhsdesign(n,p)

返回大小为 n×p 的拉丁超立方样本矩阵。对于 X 的每一列，n 值随机分布，每个区间 (0,1/n)、(1/n,2/n)、...、(1 - 1/n,1) 随机排列。

* + - 1. **输入参数**

n: 返回样本数。

p: 返回的变量数。

* + - 1. **返回值**

X：返回大小为 n×p 的拉丁超立方样本矩阵。

* + - 1. **开发方案**

这部分主要对试验设计lhsdesign函数进行说明，本函数通过对Python中pyDOE库lhsdesign ()函数进行重写而来，实现了matlab 试验设计lhsdesign函数的Julia语言实现。

* 1. **生存时间数据分析**
     1. **生存时间数据分析coxphfit函数**
        1. **调用方式**

b = coxphfit(X,T)

返回一个 p×1 向量 b，它是对预测变量 X 上观察到的响应 T 的 Cox 比例风险回归的系数估计值，其中 T 是一个 n×1 向量或一个 n×2 矩阵，X 是一个 n×p 矩阵。

* + - 1. **输入参数**

X: 预测变量。

T: 预测变量的响应值。

* + - 1. **返回值**

b：预测变量 X 上观察到的响应 T 的 Cox 比例风险回归的系数估计值向量。

* + - 1. **开发方案**

这部分主要对生存时间数据分析coxphfit函数进行说明，本函数通过对Python中lifelines库CoxPHFitter ()函数和fit()函数进行组合和重写而来，实现了matlab生存时间数据分析coxphfit函数的Julia语言实现。

* + 1. **生存时间数据分析ecdf函数**
       1. **调用方式**

[f,x] = ecdf(y)

返回使用 y 中的数据在 x 处评估的经验累积分布函数 f。

* + - 1. **输入参数**

y: 随机变量。

* + - 1. **返回值**

f: 经验累积分布函数。

x: 随机变量值。

* + - 1. **开发方案**

这部分主要对生存时间数据分析ecdf函数进行说明，本函数通过对Python中statsmodels.api.distributions库ECDF ()函数进行重写而来，实现了matlab生存时间数据分析ecdf函数的Julia语言实现。

* + 1. **生存时间数据分析ksdensity函数**
       1. **调用方式**

[f,xi] = ksdensity(x)

返回向量或两列矩阵 x 中样本数据的概率密度估计值 f。该估计基于一个正常的核函数，并在等距点 xi 处进行评估，该点覆盖了 x 中的数据范围。

* + - 1. **输入参数**

x: 向量或两列矩阵。

* + - 1. **返回值**

f: 概率密度估计值。

xi: 等距向量。

* + - 1. **开发方案**

这部分主要对生存时间数据分析ksdensity函数进行说明，本函数通过对Python中scipy.signal库savgol\_filter ()函数进行重写而来，实现了matlab生存时间数据分析ksdensity函数的Julia语言实现。

* + 1. **生存时间数据分析mle函数**
       1. **调用方式**

phat = mle(data)

使用样本数据返回正态分布参数的最大似然估计 (MLE)。

* + - 1. **输入参数**

data: 样本数据。

* + - 1. **返回值**

phat：最大似然估计值 (MLE)。

* + - 1. **开发方案**

这部分主要对生存时间数据分析mle函数进行说明，本函数通过对Python中scipy.stats.norm库fit ()函数进行重写而来，实现了matlab生存时间数据分析mle函数的Julia语言实现。

* + 1. **生存时间数据分析fitdist函数**
       1. **调用方式**

pd = fitdist(x,distname)

通过对列向量 x 中的数据进行 distname 指定的分布拟合，创建概率分布对象。

* + - 1. **输入参数**

x: 样本数据。

distname: 指定的分布。

* + - 1. **返回值**

pd：概率分布，以概率分布对象形式返回。

* + - 1. **开发方案**

这部分主要对生存时间数据分析fitdist函数进行说明，本函数通过对Python中fitter库Fitter ()函数和fit()函数进行组合和重写而来，实现了matlab生存时间数据分析fitdist函数的Julia语言实现。

* + 1. **生存时间数据分析evfit函数**
       1. **调用方式**

parmhat = evfit(data)

给定 data 中的样本数据，parmhat = evfit(data) 返回类型 1 极值分布参数的最大似然估计。样本数据 data 必须是双精度向量。 parmhat(1) 是位置参数 μ，parmhat(2) 是尺度参数 σ。

* + - 1. **输入参数**

data: 样本数据。

* + - 1. **返回值**

parmhat：分布参数值。

* + - 1. **开发方案**

这部分主要对生存时间数据分析evfit函数进行说明，本函数通过对Python中scipy.stats库genextreme ()函数和fit()函数进行组合和重写而来，实现了matlab生存时间数据分析evfit函数的Julia语言实现。

* + 1. **生存时间数据分析expfit函数**
       1. **调用方式**

muhat = expfit(data)

估计向量数据中指数分布的样本数据的平均值。

* + - 1. **输入参数**

data: 样本数据。

* + - 1. **返回值**

muhat：分布参数值。

* + - 1. **开发方案**

这部分主要对生存时间数据分析expfit函数进行说明，本函数通过对Python中scipy.stats库expon ()函数和fit()函数进行组合和重写而来，实现了matlab生存时间数据分析expfit函数的Julia语言实现。

* + 1. **生存时间数据分析gamfit函数**
       1. **调用方式**

phat = gamfit(data)

返回给定向量数据中 gamma 分布参数的最大似然估计 (MLE)。

* + - 1. **输入参数**

data: 样本数据。

* + - 1. **返回值**

phat：分布参数值。

* + - 1. **开发方案**

这部分主要对生存时间数据分析gamfit函数进行说明，本函数通过对Python中scipy.stats库gamma ()函数和fit()函数进行组合和重写而来，实现了matlab生存时间数据分析gamfit函数的Julia语言实现。

* + 1. **生存时间数据分析lognfit函数**
       1. **调用方式**

pHat = lognfit(x)

给定 x 中的样本数据，pHat = lognfit(x) 返回对数正态分布参数的无偏估计。 pHat(1) 和 pHat(2) 分别是对数值的平均值和标准差。

* + - 1. **输入参数**

x: 样本数据。

* + - 1. **返回值**

pHat：分布参数值。

* + - 1. **开发方案**

这部分主要对生存时间数据分析lognfit函数进行说明，本函数通过对Python中scipy.stats库lognorm ()函数和fit()函数进行组合和重写而来，实现了matlab生存时间数据分析lognfit函数的Julia语言实现。

* + 1. **生存时间数据分析normfit函数**
       1. **调用方式**

[muHat,sigmaHat] = normfit(x)

返回正态分布参数的估计值（均值 muHat 和标准差 sigmaHat），给定 x 中的样本数据。 muHat 是样本均值，sigmaHat 是方差的无偏估计量的平方根。

* + - 1. **输入参数**

x: 样本数据。

* + - 1. **返回值**

muHat：分布参数值。

sigmaHat：分布参数值。

* + - 1. **开发方案**

这部分主要对生存时间数据分析normfit函数进行说明，本函数通过对Python中scipy.stats库norm ()函数和fit()函数进行组合和重写而来，实现了matlab生存时间数据分析normfit函数的Julia语言实现。

* + 1. **生存时间数据分析wblfit函数**
       1. **调用方式**

parmHat = wblfit(x)

给定 x 中的样本数据，parmHat = wblfit(x) 返回 Weibull 分布参数（形状和尺度）的估计值。

* + - 1. **输入参数**

x: 样本数据。

* + - 1. **返回值**

parmHat：分布参数值。

* + - 1. **开发方案**

这部分主要对生存时间数据分析wblfit函数进行说明，本函数通过对Python中scipy.stats库exponweib ()函数和fit()函数进行组合和重写而来，实现了matlab生存时间数据分析wblfit函数的Julia语言实现。

* 1. **统计过程控制**
     1. **统计过程控制anova1函数**
        1. **调用方式**

p = anova1(y)

对样本数据 y 执行单变量方差分析并返回 p 值。 anova1 将 y 的每一列视为一个单独的组。该函数检验 y 列中的样本是从具有相同均值的总体中抽取的假设与总体均值不完全相同的备择假设。

* + - 1. **输入参数**

y: 样本数据。

* + - 1. **返回值**

p：拒绝原假设的最小显著性水平。

* + - 1. **开发方案**

这部分主要对统计过程控制anova1函数进行说明，本函数通过对Python中scipy.stats库f\_oneway ()函数进行重写而来，实现了matlab统计过程控制anova1函数的Julia语言实现。

* + 1. **统计过程控制anova2函数**
       1. **调用方式**

p = anova2(y,reps)

返回双因素 ANOVA 的 p 值，用于比较 y 中两列或多列和两列或多行观测值的均值。

* + - 1. **输入参数**

y: 样本数据。

reps: reps 是每个因子组组合的重复次数，它必须是常数，表示平衡设计。对于不平衡的设计，请使用 anovan。 anova2 函数测试列和行因子的主效应及其交互效应。要测试交互效果，reps 必须大于 1。

* + - 1. **返回值**

p：拒绝原假设的最小显著性水平。

* + - 1. **开发方案**

这部分主要对统计过程控制anova2函数进行说明，本函数通过对Python中statsmodels.stats.anova库anova\_lm ()函数进行重写而来，实现了matlab统计过程控制anova2函数的Julia语言实现。

* + 1. **统计过程控制anovan函数**
       1. **调用方式**

p = anovan(y,group)

返回一个 p 值向量，每项一个，用于多因素（n 因素）方差分析 (ANOVA)，以测试多个因素对向量 y 均值的影响。

* + - 1. **输入参数**

y: 样本数据。

group: 变量组数。

* + - 1. **返回值**

p：拒绝原假设的最小显著性水平。

* + - 1. **开发方案**

这部分主要对统计过程控制anovan函数进行说明，本函数通过对Python中statsmodels.stats.anova库anova\_lm ()函数进行重写而来，实现了matlab统计过程控制anovan函数的Julia语言实现。

* + 1. **统计过程控制friedman函数**
       1. **调用方式**

p = friedman(x,reps)

返回非参数弗里德曼检验的 p 值，以比较双向布局中的列效应。 Friedman 检验列效应都相同的原假设与它们不完全相同的替代假设。

* + - 1. **输入参数**

x: 样本数据。

reps: reps 是每个因子组组合的重复次数。

* + - 1. **返回值**

p：拒绝原假设的最小显著性水平。

* + - 1. **开发方案**

这部分主要对统计过程控制friedman函数进行说明，本函数通过对Python中scipy.stats库friedmanchisquare ()函数进行重写而来，实现了matlab统计过程控制friedman函数的Julia语言实现。

* + 1. **统计过程控制kruskalwallis函数**
       1. **调用方式**

p = kruskalwallis(x)

使用 Kruskal-Wallis 检验返回矩阵 x 的每一列中的数据来自同一分布的零假设的 p 值。另一种假设是并非所有样本都来自相同的分布。 kruskalwallis 还返回 ANOVA 表和箱线图。

* + - 1. **输入参数**

x: 样本数据。

* + - 1. **返回值**

p：拒绝原假设的最小显著性水平。

* + - 1. **开发方案**

这部分主要对统计过程控制kruskalwallis函数进行说明，本函数通过对Python中scipy.stats库kruskal ()函数进行重写而来，实现了matlab统计过程控制kruskalwallis函数的Julia语言实现。

# 绘图

## plot函数

### 调用方式

plot(X,y)

返回使用矩阵X中的预测值和向量y中的响应值所绘制的图形。

plot(y\_pred)

返回经过模型预测之后的结果图。

### 输入参数

X: 包含列中自变量值的矩阵。

y：包含因变量值的向量。

y\_pred：模型预测结果向量。

### 返回值

无

### 开发方案

这部分主要对绘图进行说明，本函数通过对Julia中Plots库plot()函数进行重写而来，实现了matlab 绘图plot函数的Julia语言实现。

## plotPartialDependence函数

### 调用方式

plotPartialDependence(RegressionMdl,Vars)

返回创建的部分依赖图（PDP）和个别条件期望图（ICE）。

### 输入参数

RegressionMdl: 训练完成的回归模型。

Vars：列出的预测变量。

### 返回值

无

### 开发方案

这部分主要对绘图进行说明，本函数通过对Python中sklearn.inspection库plot\_partial\_dependence ()函数进行重写而来，实现了matlab 绘图plotPartialDependence函数的Julia语言实现。

## plotResiduals函数

### 调用方式

plotResiduals(mdl)

广义线性回归模型的残差图。

### 输入参数

mdl: 训练完成的回归模型。

### 返回值

无

### 开发方案

这部分主要对绘图进行说明，本函数通过对Python中seaborn库residplot ()函数进行重写而来，实现了matlab 绘图plotResiduals函数的Julia语言实现。

## histfit函数

### 调用方式

histfit(data)

绘制 data 中的值的直方图并拟合密度函数，直方图的 bin 个数等于 data 中元素个数的平方根。

### 输入参数

data: 包含自变量值的向量。

### 返回值

无

### 开发方案

这部分主要对绘图进行说明，本函数通过对Python中matplotlib.pyplot库hist ()函数进行重写而来，实现了matlab 绘图 histfit函数的Julia语言实现。

# 开发计划