

快速入门指南 MATLAB 机器学习

在 Learner App 中训练模型

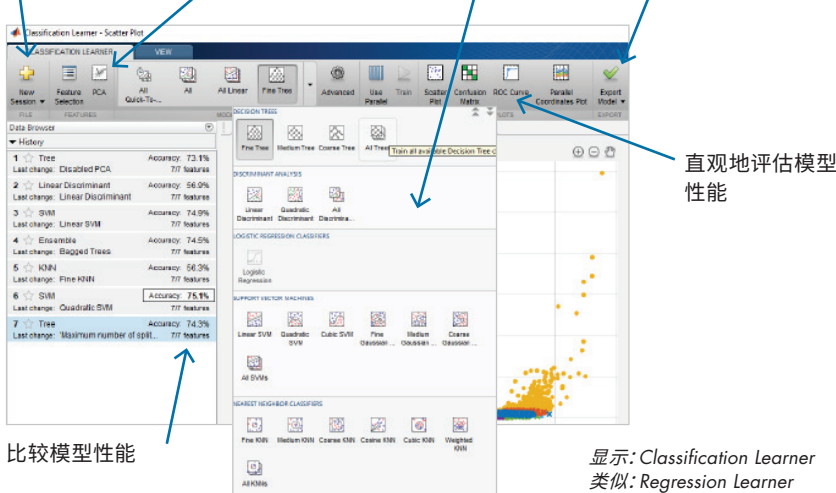
从工作区或文件中导入数据

启用 PCA 实现特征降维

从所有常见算法中选择

导出模型, 以进行进一步优化或与其他代码集成

直观地评估模型性能



比较模型性能

显示: Classification Learner 类似: Regression Learner

机器学习函数

命名约定

fit + c(classification) / r(egression) + model
例如, 对于 SVM 分类器 $m = \text{fitcsvm}(X,Y)$

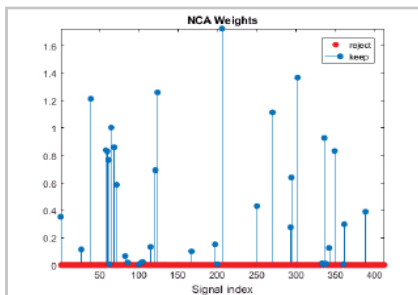
算法		模型
决策树	c,r	tree
线性模型	c,r	linear
支持向量	c,r	svm
高斯核	c,r	kernel
集成 (包括随机森林)	c,r	ensemble
K-最近邻	c	knn
判别分析	c	discr
朴素贝叶斯	c	nb
高斯过程	r	gp
(Gen.)线性回归		(g)lm
非线性回归		nlm

特征选择

近邻元分析

利用预测能力自动识别特征。

```
fscnca(X labels, 'Lambda',...);
find mdl.FeatureWeights > 0.01
```

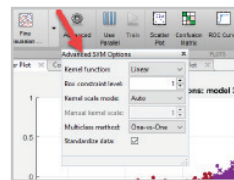


还可使用:

- 矩阵因式分解
- PCA
- 稀疏滤波
- 逐步回归
- 重建 ICA
- t-SNE

超参数调优

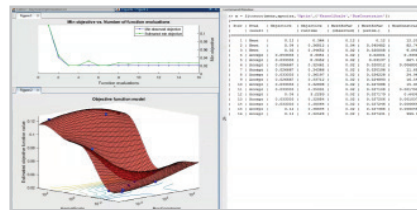
浏览并更改应用程序中的参数:



自动贝叶斯优化

利用贝叶斯模型, 决定超参数空间的哪些点进行后续尝试。比网格搜索快得多。

```
mdl = fit...(X,labels,
'OptimizeHyperparameters','auto');
```



部署

独立程序、Web 应用程序、Spark

作为独立程序、MapReduce 和 Apache Spark™ 应用程序、Web 应用程序以及 Microsoft® Excel® 插件共享。

与企业 IT/OT 集成

使用 MATLAB Compiler SDK™ 转换成 C/C++、Java®、.NET 或 Python® 库。

C 代码生成

使用 MATLAB Coder™ 自动转换为 C/C++ 代码进行嵌入式部署

1. 训练模型 $Mdl = \text{fitcsvm}(X,Y)$;
2. $\text{saveCompactModel}(Mdl, 'mySVM');$;
3. 定义入口点函数

```
function label = predictSVM(x)
    m = loadCompactModel('mySVM');
    label = predict(m,x);
end
```
4. 生成 C 代码

```
codegen predictSVM -args {X}
```

了解更多: mathworks.com/machine-learning