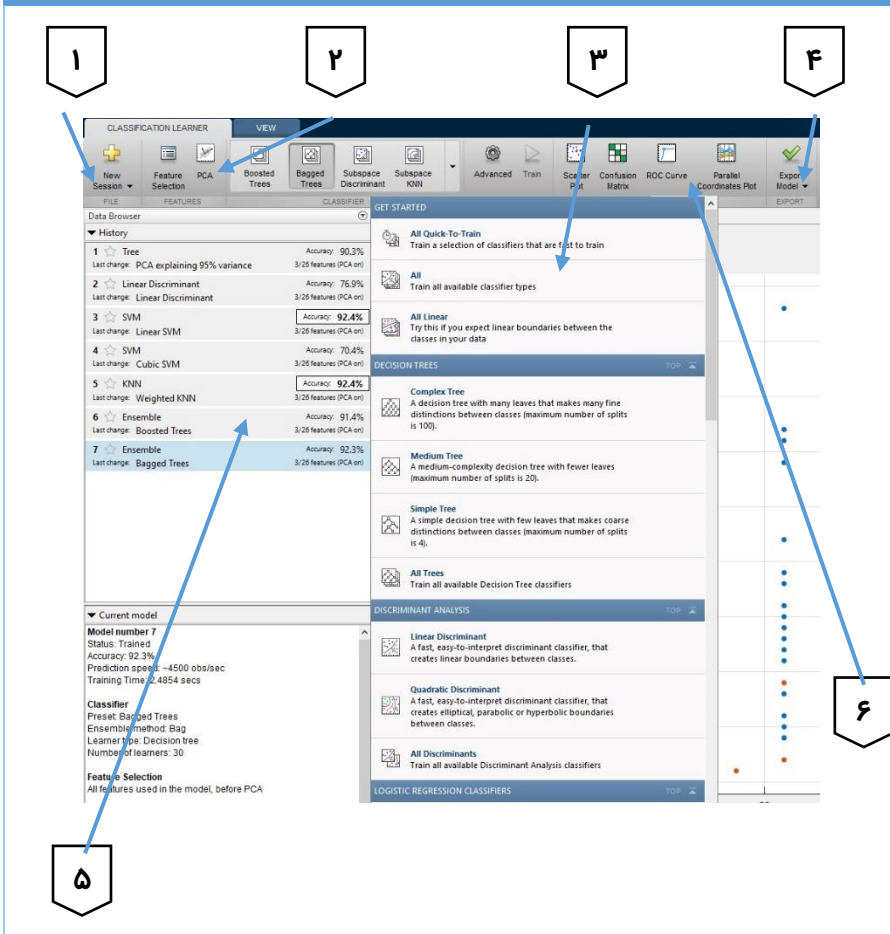


«یادگیری ماشین با متلب» (Machine Learning With MATLAB)

آموزش مدل‌های یادگیری ماشین در «ابزار یادگیری ماشین» (Learner Toolbox) متلب



۱. از طریق این گزینه، داده‌های ورودی لازم برای آموزش مدل یادگیری، از طریق فایل یا «فضای کاری» (WorkPlace) متلب وارد (Import) می‌شوند.
۲. از طریق این گزینه، روش «تحلیل مؤلفه اساسی» (Principal Component Analysis) جهت کاهش ابعاد فضای ویژگی، فعال می‌شود.
۳. از طریق این گزینه، می‌توان از میان الگوریتم‌های شناخته شده یادگیری ماشین، الگوریتم مورد نظر را برای آموزش انتخاب کرد.
۴. از طریق این گزینه، مدل یادگیری آموزش دیده شده، برای بهینه‌سازی بیشتر و یا یکپارچه‌سازی با دیگر کدها، در فضای کاری متلب قرار می‌گیرد.
۵. در این قسمت، کاربر می‌تواند مدل‌های یادگیری ماشین آموزش دیده را مشاهده و عملکرد آن‌ها را با یکدیگر مقایسه کند.
۶. از طریق گزینه‌های موجود در این بخش، می‌توان به صورت بصری عملکرد مدل‌های یادگیری ماشین آموزش دیده را ارزیابی کرد.



توابع یادگیری ماشین

قواعد نام‌گذاری توابع آموزش مدل‌های یادگیری ماشین در متلب

برای نام‌گذاری توابع یادگیری ماشین در زبان برنامه‌نویسی متلب، از نام‌های سه بخشی استفاده می‌شود. نام‌ها با کلمه fit آغاز می‌شوند. بلافاصله پس از آن، در صورتی که مدل یادگیری از نوع «دسته‌بندی» (Classification) باشد، از حرف c و در صورتی که از نوع «رگرسیون» (Regression) باشد، از حرف r استفاده می‌شود. سپس، نام تعریف شده برای مدل یادگیری ماشین، در متلب آورده می‌شود. به عنوان نمونه، برای یک دسته‌بند «ماشین بردار پشتیبان» (Support Vector Machine) از کد زیر استفاده می‌شود:

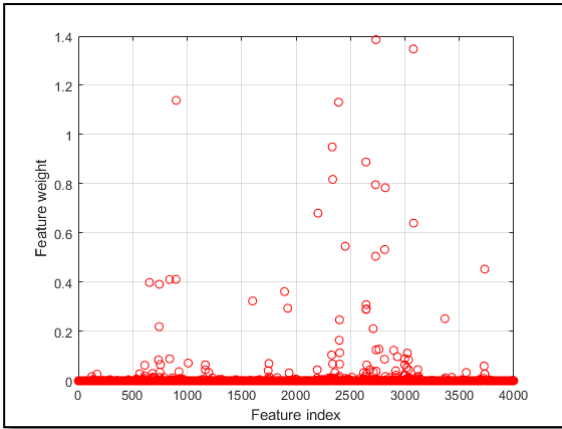
$$m = fitcsvm (X,Y)$$

نام تعریف شده	دسته‌بندی / رگرسیون	الگوریتم یادگیری	نام تعریف شده	دسته‌بندی / رگرسیون	الگوریتم یادگیری
discr	دسته‌بندی	«دسته‌بندی تحلیل جداپذیری» (Discrim. Analysis)	tree	دسته‌بندی و رگرسیون	«درخت تصمیم» (Decision Tree)
nb	دسته‌بندی	«روش بیز» (Naïve Bayes)	linear	دسته‌بندی و رگرسیون	«مدل خطی» (Linear Model)
gp	رگرسیون	«فرآیندهای گوسی» (Gaussian Process)	svm	دسته‌بندی و رگرسیون	«ماشین بردار پشتیبان» (SVM)
(g)lm	رگرسیون	«رگرسیون خطی» (Linear regression)	kernel	دسته‌بندی و رگرسیون	«کرنل گوسی» (Gaussian kernel)
nlm	رگرسیون	«رگرسیون غیر خطی» (Non-Linear Regression)	ensemble	دسته‌بندی و رگرسیون	«مدل‌های ترکیبی» (Ensembles)
			klmn	دسته‌بندی و رگرسیون	«k-نزدیک‌ترین همسایه» (KNN)

انتخاب ویژگی

«تحلیل مؤلفه همسایگی» (Neighborhood Component Analysis)

با استفاده از الگوریتم‌های انتخاب ویژگی موجود در متلب (نظیر تحلیل مؤلفه همسایگی)، این امکان به راحتی فراهم می‌شود تا فرآیند شناسایی ویژگی‌های مطلوب برای دسته‌بندی و رگرسیون را خودکار کرد. ویژگی مهم چنین الگوریتم‌هایی، عملکرد بالای آن‌ها در پیش‌بینی و انتخاب ویژگی‌هایی است که عملکرد روش‌های یادگیری ماشین را بهبود می‌بخشد.



نمونه کد انتخاب ویژگی برای دسته‌بندی

```
mdl = fscnca(X,y,'Solver','sgd','Verbose',1);
plot(mdl.FeatureWeights,'ro');
find(mdl.FeatureWeights > 0.01);
```

نمونه کد انتخاب ویژگی برای رگرسیون

```
mdl = fsrnca(X,y,'Verbose',1,'Lambda',0.5/N);
plot(mdl.FeatureWeights,'ro');
find(mdl.FeatureWeights > 0.01);
```

دیگر الگوریتم‌های انتخاب ویژگی پیاده‌سازی شده در متلب

۱. تحلیل مؤلفه اساسی (Principal Component Analysis | PCA)
۲. پالایش اسپارس (Sparse Filtering)
۳. فاکتورگیری ماتریس (Matrix Factorization)
۴. رگرسیون گام به گام (Stepwise Regression)
۵. بازسازی تحلیل مؤلفه مستقل (Reconstruction ICA)

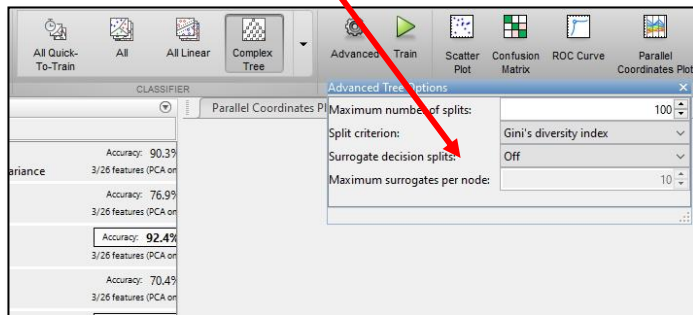
TUNING ITER	TUNING SUBSET FUN VALUE	LEARNING RATE
1	2.403497e+01	2.000000e-01
2	2.275050e+01	4.000000e-01
3	2.036845e+01	8.000000e-01
4	1.627647e+01	1.600000e+00
5	1.023512e+01	3.200000e+00
6	3.864283e+00	6.400000e+00
7	4.743816e-01	1.280000e+01
8	-7.260138e-01	2.560000e+01
9	-7.260138e-01	2.560000e+01
10	-7.260138e-01	2.560000e+01
11	-7.260138e-01	2.560000e+01
12	-7.260138e-01	2.560000e+01
13	-7.260138e-01	2.560000e+01
14	-7.260138e-01	2.560000e+01
15	-7.260138e-01	2.560000e+01
16	-7.260138e-01	2.560000e+01
17	-7.260138e-01	2.560000e+01
18	-7.260138e-01	2.560000e+01
19	-7.260138e-01	2.560000e+01
20	-7.260138e-01	2.560000e+01

o Solver = SGD, MiniBatchSize = 10, PassLimit = 5

PASS	ITER	AVG MINIBATCH FUN VALUE	AVG MINIBATCH NORM GRAD	NORM STEP	LEARNING RATE
0	9	4.016978e+00	2.835465e-02	5.395984e+00	2.560000e+01
1	19	-6.726156e-01	6.111354e-02	5.021138e-01	1.280000e+01
1	29	-8.316555e-01	4.024185e-02	1.196030e+00	1.280000e+01
2	39	-8.838656e-01	2.333418e-02	1.225839e-01	8.533333e+00
3	49	-8.669035e-01	3.413150e-02	3.421881e-01	6.400000e+00
3	59	-8.906935e-01	1.946293e-02	2.232510e-01	6.400000e+00
4	69	-8.778630e-01	3.561283e-02	3.290643e-01	5.120000e+00
4	79	-8.857136e-01	2.516633e-02	3.902977e-01	5.120000e+00

«میزان سازی ابرپارامترها» (Hyperparameter Tuning)

کاربر می تواند پارامترها را در ابزار یادگیری ماشین متلب مشاهده کرده و تغییر دهد.

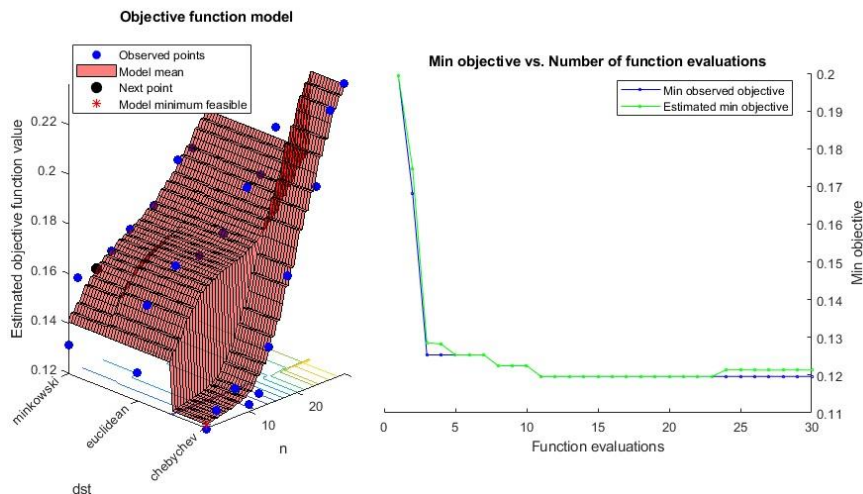


با استفاده از ابزارهای یادگیری ماشین متلب، این امکان برای کاربران فراهم شده است تا ابرپارامترهای بهینه برای مدل های یادگیری ماشین را با استفاده از روش «بهینه سازی بیزی» (Bayesian Optimization) به طور خودکار انتخاب و برای آموزش مدل یادگیری ماشین استفاده کنند.

در ابزارهای یادگیری ماشین، از مدل بیزی استفاده می شود تا مشخص شود کدام ویژگی ها در فضای ابرپارامتری، تاثیر بیشتری بر عملکرد و دقت مدل یادگیری ماشین خواهند داشت. چنین ساختاری برای انتخاب ویژگی، به مراتب سریع تر از روش هایی نظیر جستجوی شبکه ای (Grid Search) است.

نمونه کد میزان سازی ابرپارامترها برای دسته بند K-نزدیک ترین همسایه

```
load ionosphere
num = optimizableVariable('n', [1,30], 'Type', 'integer');
dst = optimizableVariable('dst', {'chebychev', 'euclidean', 'minkowski'}, 'Type', 'categorical');
c = cvpartition(351, 'Kfold', 5);
fun = @(x) kfoldLoss(fitcknn(X,Y, 'CVPartition', c, 'NumNeighbors', x.n, ...
    'Distance', char(x.dst), 'NSMethod', 'exhaustive'));
results = bayesopt(fun, [num,dst], 'Verbose', 0, ...
    'AcquisitionFunctionName', 'expected-improvement-plus');
```





به کاراندازی مدل‌های یادگیری ماشین (Deploy)

به کاراندازی در کاربردهای «مستقل» (Standalone)، «نرم‌افزارهای تحت وب» (Web Apps) و کتابخانه نرم‌افزاری Spark

مدل‌های یادگیری ماشین آموزش دیده شده در متلب را به راحتی می‌توان به عنوان یک نرم‌افزار مستقل (برای کاربردهای مختلف)، مدل‌های قابل استفاده در کاربردهای مبتنی بر کتابخانه‌های نرم‌افزاری Spark و MapReduce (توسعه داده شده توسط «بنیاد نرم‌افزاری Apache» (Apache Software Foundation))، نرم‌افزارهای تحت وب و افزونه‌های نرم‌افزار اکسل (Microsoft Excel add-ins) به کار گرفت.

یکپارچه‌سازی با دیگر پلتفرم‌های توسعه نرم‌افزار

کدهای نوشته شده به زبان برنامه‌نویسی متلب را می‌توان با استفاده از «کیت توسعه نرم‌افزار» (Software Development Kit) کامپایلر متلب، به کتابخانه‌های نرم‌افزاری زبان‌های برنامه‌نویسی دیگر نظیر «سی» (C)، «سی پلاس پلاس» (C++)، «جاوا» (Java)، «دات‌نت» (.NET) یا «پایتون» (Python) تبدیل کرد.

تولید کدهای C از طریق تبدیل کدهای نوشته شده در متلب

از طریق قابلیت کاربردی تعبیه شده در نرم‌افزار متلب به نام MATLAB Coder، کدهای نوشته شده در متلب را می‌توان به کدهای زبان C یا C++ تبدیل کرد. کدهای تبدیل شده را می‌توان برای یکپارچه‌سازی با کدهای دیگر، به کار انداخت.

۱. ابتدا، مدل یادگیری ماشین، آموزش داده می‌شود.

```
Mdl = fitcsvm(X,Y);  
saveCompactModel(Mdl, 'mySVM');
```



۲. سپس، تابع «نقطه ورود» (Entry-Point) برای مدل یادگیری ماشین آموزش داده شده تعریف می‌شود.

```
function label = predictSVM(x)
    m = loadCompactModel('mySVM');
    label = predict(m,x);
end
```

۳. در نهایت، دستور تولید کد C یا C++ داده می‌شود.

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

مجموعه آموزش‌های الگوریتم‌های داده‌کاوی و یادگیری ماشین (+کلیک کنید)

برای مشاهده دیگر «تقلب‌نامه‌های» مجله فرادرس، به [این لینک](#) مراجعه فرمایید.

جهت آگاهی از آخرین تقلب‌نامه‌های منتشر شده، در [کانال تلگرام](#) مجله فرادرس عضو شوید.

تهیه و تنظیم: مجله فرادرس

