

# 算法测试

特征提取的话，参照自然语言处理，把命令当做文档向量化，然后使用sklearn中的算法进行分类，类似文本情感极性分析，目前只考虑了命令，这一单个因素。使用knn、svm、决策树与kmeans算法进行了测试，knn由于效果最差pass掉了。由于恶意数据集特别小，只有234条指令，训练的时候是从正常数据集中随机取1000条，与恶意数据混合后进行训练与测试。结果表明，决策树在分类方面具有较好的准确性，测试的表现更好。

使用单一样本建模，使用聚类的方式进行恶意命令识别，测试了kmeans算法性能，结果一般，目前正在优化该类型的算法在此问题中的应用。

不管哪种方法，准确性距离要求的还有距离。未来的目标是复现论文Detection of Malicious Remote Shell Sessions中的分类系统。

## svm结果分析

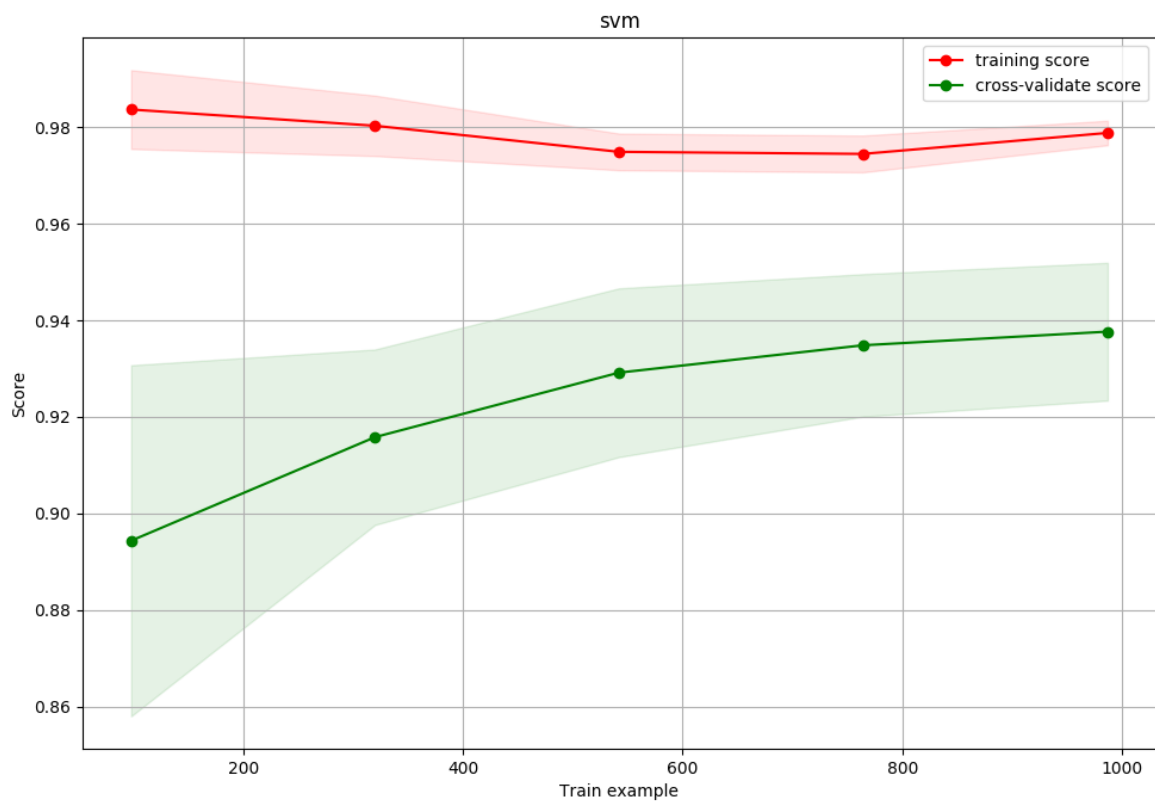
在随机数据集上进行参数选择测试，最优参数为 rbf，gamma取值0.22449

```
clf = svm.SVC(kernel='rbf', gamma=0.22449)
```

在某次测试中得到结果如下

2020-06-08 11:10:57,994	train.py 215: INFO	report for svm:		
2020-06-08 11:11:01,955	train.py 221: INFO	svm score: 0.9473684210526315		
	precision	recall	f1-score	support
0	0.97	0.97	0.97	203
1	0.86	0.84	0.85	44
accuracy			0.95	247
macro avg	0.91	0.91	0.91	247
weighted avg	0.95	0.95	0.95	247
2020-06-08 11:11:02,473	train.py 190: INFO	test decision tree:		
		[0.93117409 0.94331984 0.93522267 0.95546559 0.93089431]		

学习曲线如下



kfold 5折交叉验证

[0.93117409 0.94331984 0.93522267 0.95546559 0.93089431]

从学习曲线看出，准确率未收敛，感觉有些欠拟合，恶意数据有些少，导致模型训练结果不理想

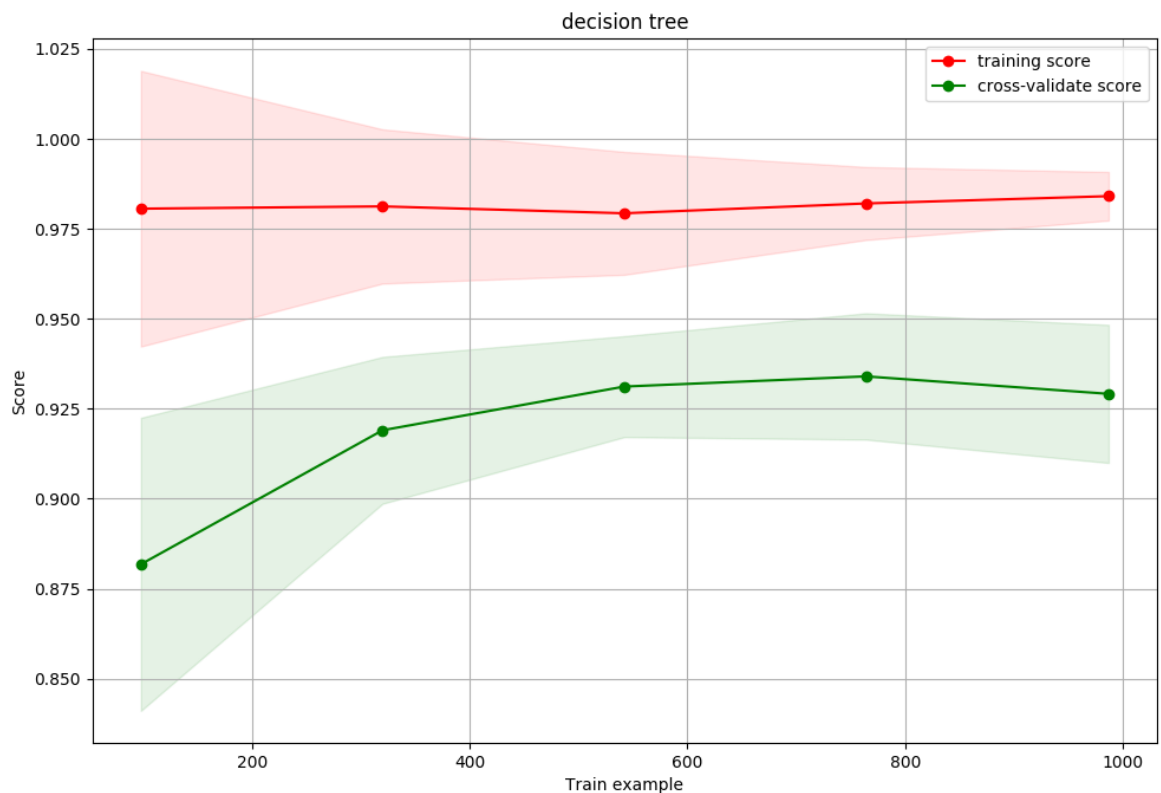
## decision 结果分析

decision tree结果

```
2020-06-08 11:11:05,592 train.py 206: INFO best decision tree param: {'min_samples_split': 14}
best score: 0.9367909238249594
2020-06-08 11:11:05,593 train.py 209: INFO decision tree score: 0.979757085020243
2020-06-08 11:11:05,595 train.py 210: INFO
```

	precision	recall	f1-score	support
0	0.99	0.98	0.99	199
1	0.94	0.96	0.95	48
accuracy			0.98	247
macro avg	0.96	0.97	0.97	247
weighted avg	0.98	0.98	0.98	247

学习曲线



kfold 5折交叉验证

**[0.93927126 0.94736842 0.92307692 0.93117409 0.93495935]**

与svm相比，决策树训练过程表现一般，但测试过程表现良好（对新数据的测试准确率更高），而且对恶意数据的准确性较svm高

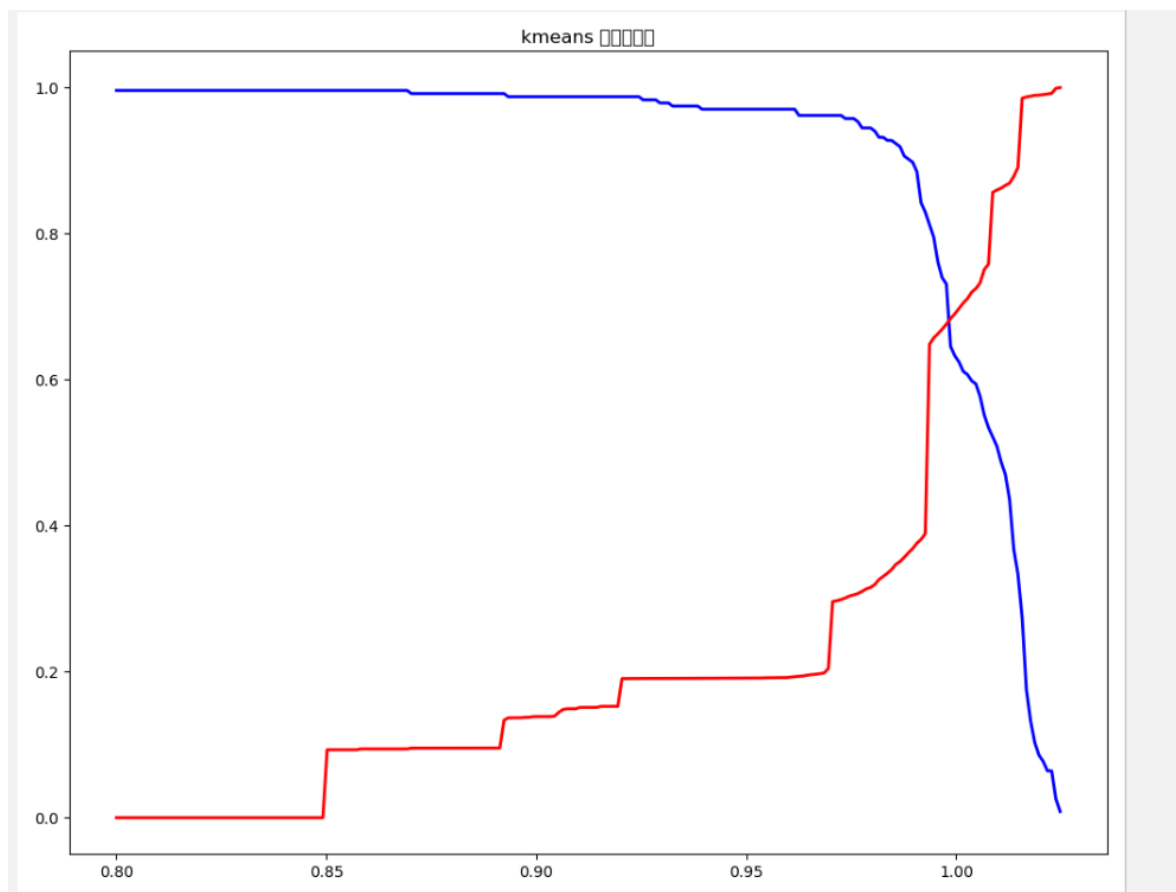
0	0.99	0.98	0.99	199
1	0.94	0.96	0.95	48

但是，学习曲线已经出现了下降趋势，表示模型已收敛，上限一般。使用训练出的模型，对所有数据进行测试分类，恶意数据准确性出奇的低

2020-07-01 19:52:04,269	train.py 321: INFO	decision tree score: 0.7757051186390092			
2020-07-01 19:52:04,326	train.py 322: INFO	precision	recall	f1-score	support
0	1.00	0.77	0.87		39972
1	0.02	0.98	0.05		234
accuracy			0.78		40206
macro avg	0.51	0.88	0.46		40206
weighted avg	0.99	0.78	0.87		40206

## kmeans分析

换一种思路，由于恶意数据集非常小，可以拿正常数据建模，得到聚类中心，然后通过计算样本命令与聚类中心的距离，判断命令是否恶意，需要设置合适的阈值。所以使用kmeans算法训练模型，使用所有数据一起测试



红色是正常数据准确率，蓝色是异常数据准确率，横轴是选择的阈值，随着阈值提高，曲线变化在大概 1.00 时达到折中的一点，此时准确性大概 67%。效果不是很理想。

## 结论

算法优化是一方面，最主要的问题在特征提取。如果可以综合考虑更多的特征，应该可以使得模型准确性进一步提高。未来的方向是复现 Detection of Malicious Remote Shell Sessions 里的分类算法。