

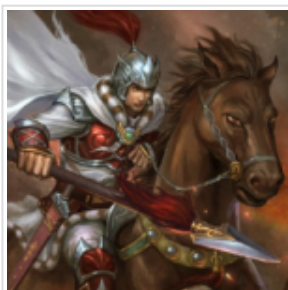
## qinhanmin2010的博客

目录视图

摘要视图

RSS 订阅

## 个人资料



qinhanmin

访问：10979次

积分：373

等级：BLOG &gt; 2

排名：千里之外

原创：4篇 转载：1篇

译文：23篇 评论：0条

## 文章搜索

异步赠书：9月重磅新书升级，本本经典

程序员9月书讯

每周荐书：ES6、虚拟现实、物联网（评论送书）

## Machine Learning with Scikit-Learn and Tensorflow 6.9 决策树局限性

2017-04-01 11:37

325人阅读

评论

分类：机器学习（25）

## 书籍信息

Hands-On Machine Learning with Scikit-Learn and Tensorflow

出版社: O'Reilly Media, Inc, USA

平装: 566页

语种: 英语

ISBN: 1491962291

条形码: 9781491962299

商品尺寸: 18 x 2.9 x 23.3 cm

ASIN: 1491962291

系列博文为书籍中文翻译

代码以及数据下载：<https://github.com/ageron/handson-ml>

决策树存在部分局限性。首先，决策树倾向水平/垂直的决策边界，使得决策树对数据的旋转敏感。以下图为例，右边的数据由左边的数据旋转45度得到，显然左边的模型优于右边的模型。解决这样的问题的方法包括通过PCA进

关闭

## 文章分类

机器学习 (26)

python基础 (3)

开发攻略 (0)

## 文章存档

2017年10月 (3)

2017年09月 (1)

2017年04月 (17)

2017年03月 (9)

## 阅读排行

Machine Learning with S (1608)

python机器学习模型选择 (1369)

Machine Learning with S (1275)

Machine Learning with S (1236)

Machine Learning with S (602)

Machine Learning with S (411)

Machine Learning with S (365)

Machine Learning with S (349)

Machine Learning with S (346)

Machine Learning with S (325)

## 评论排行

Notepad++拼写检查插件 (0)

Machine Learning with S (0)

行降维。

```

1  # create the data
2  rnd.seed(6)
3  Xs = rnd.rand(100, 2) - 0.5
4  ys = (Xs[:, 0] > 0).astype(np.float32)
5
6  # do the rotation
7  angle = np.pi / 4
8  rotation_matrix = np.array([[np.cos(angle), -np.sin(angle)], [np.sin(angle), np.cos(angle)]])
9  Xsr = Xs.dot(rotation_matrix)
10
11 # train the model
12 tree_clf_s = DecisionTreeClassifier(random_state=42)
13 tree_clf_s.fit(Xs, ys)
14 tree_clf_sr = DecisionTreeClassifier(random_state=42)
15 tree_clf_sr.fit(Xsr, ys)
16
17 # visualize the model
18 def plot_decision_boundary(clf, X, y, axes):
19     x1s = np.linspace(axes[0], axes[1], 100)
20     x2s = np.linspace(axes[2], axes[3], 100)
21     x1, x2 = np.meshgrid(x1s, x2s)
22     X_new = np.c_[x1.ravel(), x2.ravel()]
23     y_pred = clf.predict(X_new).reshape(x1.shape)
24     custom_cmap = ListedColormap(['#fafab0', '#a0faa0'])
25     plt.contourf(x1, x2, y_pred, alpha=0.3, cmap=custom_cmap, linewidth=10)
26     plt.plot(X[:, 0][y==0], X[:, 1][y==0], "yo")
27     plt.plot(X[:, 0][y==1], X[:, 1][y==1], "g^")
28     plt.axis(axes)
29     plt.xlabel(r"$x_1$", fontsize=18)
30     plt.ylabel(r"$x_2$", fontsize=18, rotation=0)
31
32 plt.figure(figsize=(11, 4))
33 plt.subplot(121)

```

关闭

Machine Learning with S (0)

Machine Learning with S (0)

Machine Learning with S (0)

Machine Learning with S (0)

Machine Learning with S (0)

Machine Learning with S (0)

scikit-learn分类问题入门 (0)

matplotlib绘制决策边界 (0)

#### 推荐文章

\* CSDN新版博客feed流内测用户征集令

\* Android检查更新下载安装

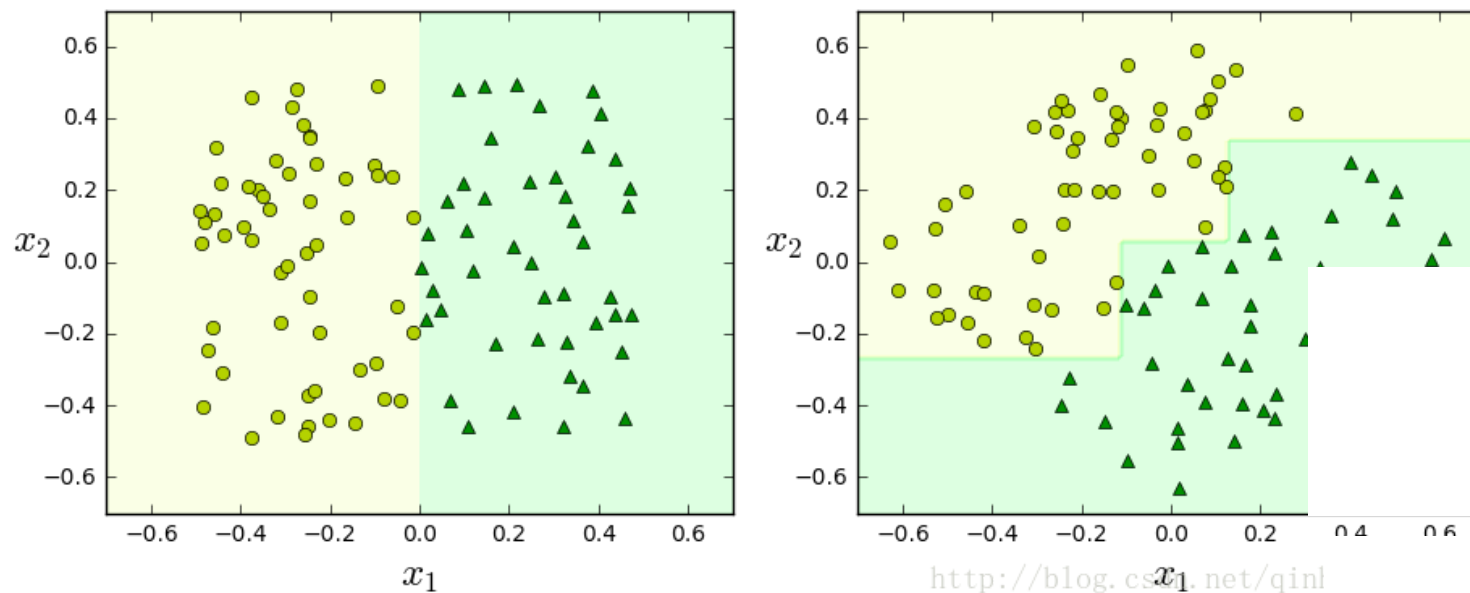
\* 动手打造史上最简单的Recycleview 侧滑菜单

\* TCP网络通讯如何解决分包粘包问题

\* SDCC 2017之大数据技术实战线上峰会

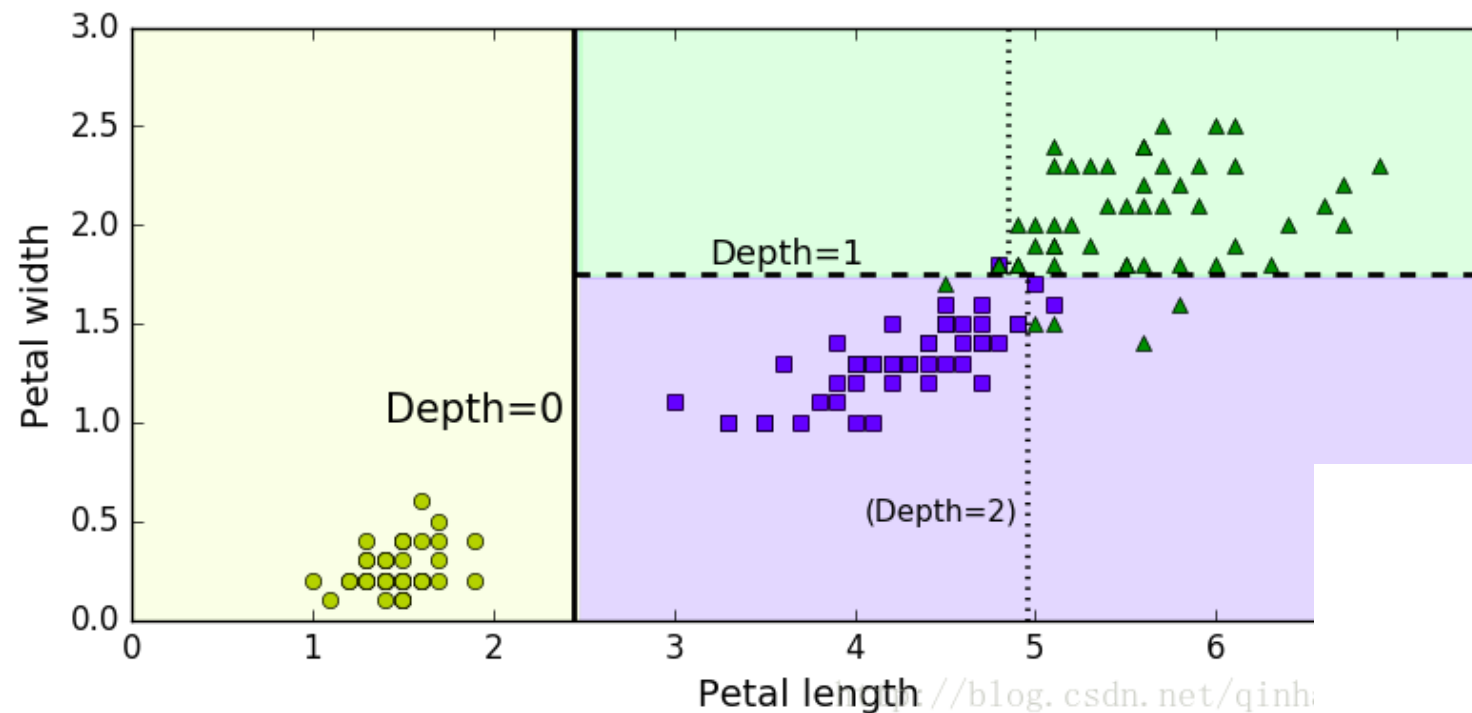
\* 快速集成一个视频直播功能

```
34 plot_decision_boundary(tree_clf_s, Xs, ys, [-0.7, 0.7, -0.7, 0.7])
35 plt.subplot(122)
36 plot_decision_boundary(tree_clf_sr, Xsr, ys, [-0.7, 0.7, -0.7, 0.7])
37 plt.show()
```



更一般地说，决策树的主要问题是面对训练数据的细微变化非常敏感。例如，如果我们移除训练数据集中的某些数据点，那么我们将得到完全不同的决策树（上面是6.2得到的决策树，下面是新的决策树）。因此，对于相同的数据集我们也有可能得到不同的结果（如果需要得到相同的结果，需要设置random\_state参数）。

关闭



```

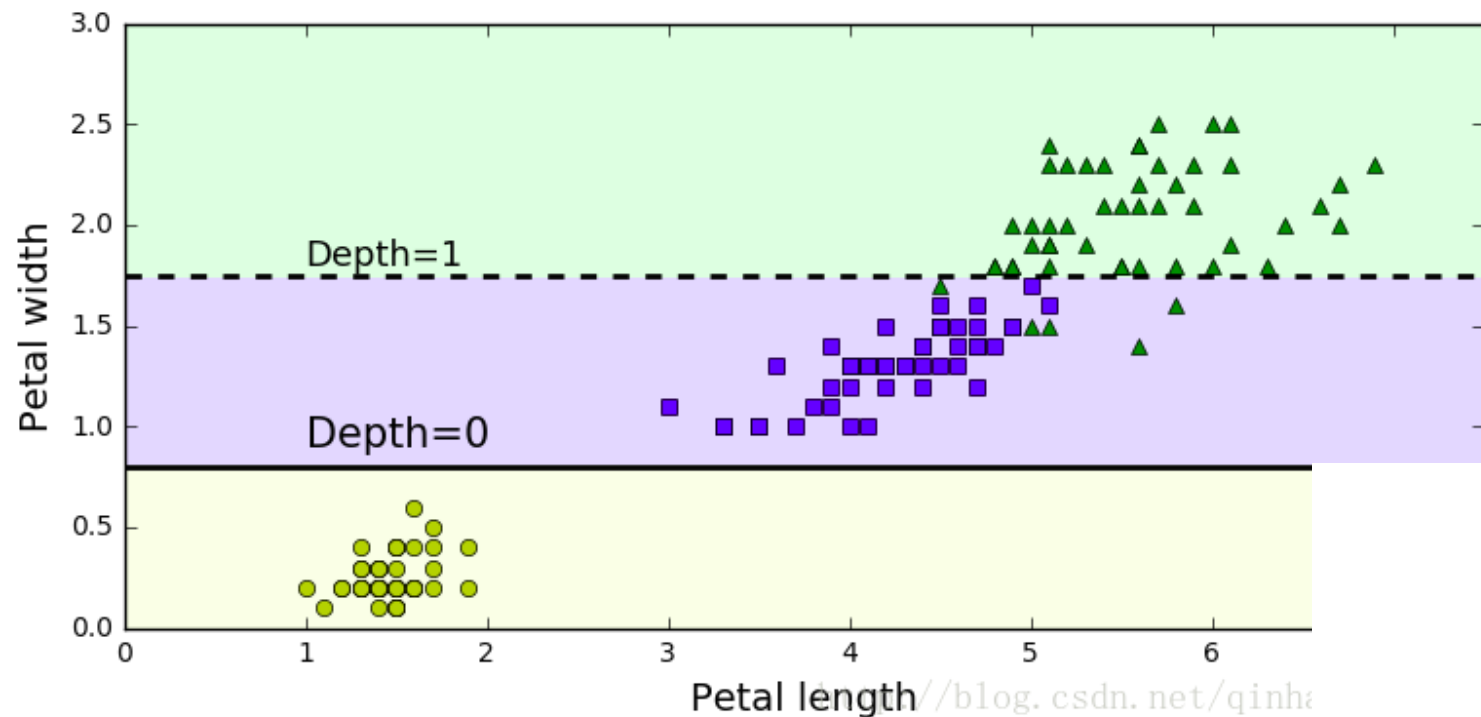
1  # load the data
2  iris = load_iris()
3  X = iris.data[:, 2:] # petal length and width
4  y = iris.target
5
6  # remove certain point
7  not_widest_versicolor = (X[:, 1] != 1.8) | (y == 2)
8  X_tweaked = X[not_widest_versicolor]
9  y_tweaked = y[not_widest_versicolor]
10
11 # train the model
12 tree_clf_tweaked = DecisionTreeClassifier(max_depth=2, random_state=40)
13 tree_clf_tweaked.fit(X_tweaked, y_tweaked)
14
15 # visualize the model
16 from matplotlib.colors import ListedColormap

```

关闭

```
17 plt.figure(figsize=(9, 4))
18 x1s = np.linspace(0, 7.5, 100)
19 x2s = np.linspace(0, 3, 100)
20 x1, x2 = np.meshgrid(x1s, x2s)
21 X_new = np.c_[x1.ravel(), x2.ravel()]
22 y_pred = tree_clf_tweaked.predict(X_new).reshape(x1.shape)
23 custom_cmap = ListedColormap(['#fafab0', '#9898ff', '#a0faa0'])
24 plt.contourf(x1, x2, y_pred, alpha=0.3, cmap=custom_cmap)
25 plt.plot(X_tweaked[:, 0][y_tweaked==0], X_tweaked[:, 1][y_tweaked==0], "yo", label="Iris-Setosa")
26 plt.plot(X_tweaked[:, 0][y_tweaked==1], X_tweaked[:, 1][y_tweaked==1], "bs", label="Iris-Versicolor")
27 plt.plot(X_tweaked[:, 0][y_tweaked==2], X_tweaked[:, 1][y_tweaked==2], "g^", label="Iris-Virginica")
28 plt.axis([0, 7.5, 0, 3])
29 plt.xlabel("Petal length", fontsize=14)
30 plt.ylabel("Petal width", fontsize=14)
31 plt.plot([0, 7.5], [0.8, 0.8], "k-", linewidth=2)
32 plt.plot([0, 7.5], [1.75, 1.75], "k--", linewidth=2)
33 plt.text(1.0, 0.9, "Depth=0", fontsize=15)
34 plt.text(1.0, 1.80, "Depth=1", fontsize=13)
35 plt.show()
```

关闭



译者注：

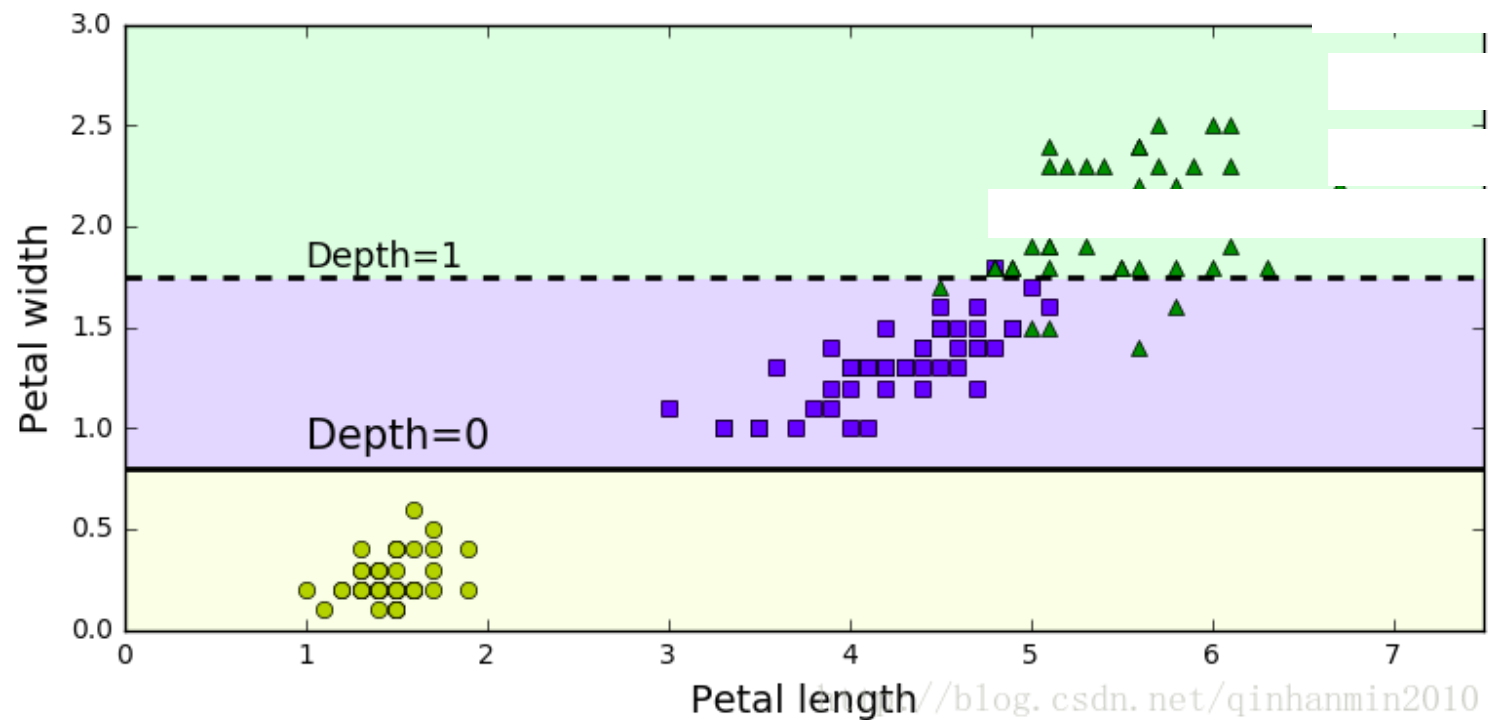
经过试验，这里结果差异的原因random\_state参数，而不是移除的特定点。注意到6.1训练决策树时设定random\_state=42，此处设定的参数是random\_state=40，如果6.1训练决策树时的结果。

```
1 iris = load_iris()
2 X = iris.data[:, 2:] # petal length and width
3 y = iris.target
4
5 tree_clf = DecisionTreeClassifier(max_depth=2, random_state=40)
6 tree_clf.fit(X, y)
7
8 plt.figure(figsize=(9, 4))
9 x1s = np.linspace(0, 7.5, 100)
10 x2s = np.linspace(0, 3, 100)
```

```

11 x1, x2 = np.meshgrid(x1s, x2s)
12 X_new = np.c_[x1.ravel(), x2.ravel()]
13 y_pred = tree_clf.predict(X_new).reshape(x1.shape)
14 custom_cmap = ListedColormap(['#fafab0', '#9898ff', '#a0faa0'])
15 plt.contourf(x1, x2, y_pred, alpha=0.3, cmap=custom_cmap)
16 plt.plot(X[:, 0][y==0], X[:, 1][y==0], "yo", label="Iris-Setosa")
17 plt.plot(X[:, 0][y==1], X[:, 1][y==1], "bs", label="Iris-Versicolor")
18 plt.plot(X[:, 0][y==2], X[:, 1][y==2], "g^", label="Iris-Virginica")
19 plt.axis([0, 7.5, 0, 3])
20 plt.xlabel("Petal length", fontsize=14)
21 plt.ylabel("Petal width", fontsize=14)
22 plt.plot([0, 7.5], [0.8, 0.8], "k-", linewidth=2)
23 plt.plot([0, 7.5], [1.75, 1.75], "k--", linewidth=2)
24 plt.text(1.0, 0.9, "Depth=0", fontsize=15)
25 plt.text(1.0, 1.80, "Depth=1", fontsize=13)
26 plt.show()

```



关闭

随机森林等集成学习方法可以通过训练多棵树降低决策树的不确定性。

顶 0 踩 0

上一篇 Machine Learning with Scikit-Learn and Tensorflow 6.8 决策树回归

下一篇 Machine Learning with Scikit-Learn and Tensorflow 6.10 练习

#### 相关文章推荐

- Machine Learning with Scikit-Learn and Tensorflow...
- Presto的服务治理与架构在京东的实践与应用--王...
- Machine Learning with Scikit-Learn and Tensorflow...
- 深入掌握Kubernetes应用实践--王渊命
- pImpl的原理
- Python基础知识汇总
- Machine Learning with Scikit-Learn and Tensorflo...
- Android核心技术详解
- Machine Learning with Scikit-Learn and Tensorfl...
- Retrofit 从入门封装到源码解析
- TensorFlow及深
- 自然语言处理工具Word2Vec
- 数十种TensorFlow实现案例汇集：代码+笔记
- Machine Learning with Scikit-Learn and Tensorflo...
- Hands-On Machine Learning with Scikit-Learn an...
- Hands-On Machine Learning with Scikit-Learn an...

查看评论

暂无评论



您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

网站客服

杂志客服

微博客服

[webmaster@csdn.net](mailto:webmaster@csdn.net)

400-660-0108

| 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 | 江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved



关闭