

23 | SVM（下）：如何进行乳腺癌检测？

2019-02-04 陈旸

数据分析实战45讲

[进入课程 >](#)



讲述：陈旸

时长 10:21 大小 23.71M



讲完了 SVM 的原理之后，今天我来带你进行 SVM 的实战。

在此之前我们先来回顾一下 SVM 的相关知识点。SVM 是有监督的学习模型，我们需要事先对数据打上分类标签，通过求解最大分类间隔来求解二分类问题。如果要求解多分类问题，可以将多个二分类器组合起来形成一个多分类器。

上一节中讲到了硬间隔、软间隔、非线性 SVM，以及分类间隔的公式，你可能会觉得比较抽象。这节课，我们会在实际使用中，讲解对工具的使用，以及相关参数的含义。

如何在 sklearn 中使用 SVM

在 Python 的 sklearn 工具包中有 SVM 算法，首先需要引用工具包：

```
1 from sklearn import svm
```

SVM 既可以做回归，也可以做分类器。

当用 SVM 做回归的时候，我们可以使用 SVR 或 LinearSVR。SVR 的英文是 Support Vector Regression。这篇文章只讲分类，这里只是简单地提一下。

当做分类器的时候，我们使用的是 SVC 或者 LinearSVC。SVC 的英文是 Support Vector Classification。

我简单说一下这两者之前的差别。

从名字上你能看出 LinearSVC 是个线性分类器，用于处理线性可分的数据，只能使用线性核函数。上一节，我讲到 SVM 是通过核函数将样本从原始空间映射到一个更高维的特质空间中，这样就使得样本在新的空间中线性可分。

如果是针对非线性的数据，需要用到 SVC。在 SVC 中，我们既可以使用到线性核函数（进行线性划分），也能使用高维的核函数（进行非线性划分）。

如何创建一个 SVM 分类器呢？

我们首先使用 SVC 的构造函数：`model = svm.SVC(kernel= 'rbf' , C=1.0, gamma= 'auto')`，这里有三个重要的参数 `kernel`、`C` 和 `gamma`。

`kernel` 代表核函数的选择，它有四种选择，只不过默认是 `rbf`，即高斯核函数。

1. `linear`: 线性核函数
2. `poly`: 多项式核函数
3. `rbf`: 高斯核函数（默认）
4. `sigmoid`: sigmoid 核函数

这四种函数代表不同的映射方式，你可能会问，在实际工作中，如何选择这 4 种核函数呢？我来给你解释一下：

线性核函数，是在数据线性可分的情况下使用的，运算速度快，效果好。不足在于它不能处理线性不可分的数据。

多项式核函数可以将数据从低维空间映射到高维空间，但参数比较多，计算量大。

高斯核函数同样可以将样本映射到高维空间，但相比于多项式核函数来说所需的参数比较少，通常性能不错，所以是默认使用的核函数。

了解深度学习的同学应该知道 sigmoid 经常用在神经网络的映射中。因此当选用 sigmoid 核函数时，SVM 实现的是多层神经网络。

上面介绍的 4 种核函数，除了第一种线性核函数外，其余 3 种都可以处理线性不可分的数据。

参数 C 代表目标函数的惩罚系数，惩罚系数指的是分错样本时的惩罚程度，默认情况下为 1.0。当 C 越大的时候，分类器的准确性越高，但同样容错率会越低，泛化能力会变差。相反，C 越小，泛化能力越强，但是准确性会降低。

参数 gamma 代表核函数的系数，默认为样本特征数的倒数，即 $\gamma = 1 / n_features$ 。

在创建 SVM 分类器之后，就可以输入训练集对它进行训练。我们使用 `model.fit(train_X, train_y)`，传入训练集中的特征值矩阵 `train_X` 和分类标识 `train_y`。特征值矩阵就是我们在特征选择后抽取的特征值矩阵（当然你也可以用全部数据作为特征值矩阵）；分类标识就是人工事先针对每个样本标识的分类结果。这样模型会自动进行分类器的训练。我们可以使用 `prediction=model.predict(test_X)` 来对结果进行预测，传入测试集中的样本特征矩阵 `test_X`，可以得到测试集的预测分类结果 `prediction`。

同样我们也可以创建线性 SVM 分类器，使用 `model=svm.LinearSVC()`。在 `LinearSVC` 中没有 `kernel` 这个参数，限制我们只能使用线性核函数。由于 `LinearSVC` 对线性分类做了优化，对于数据量大的线性可分问题，使用 `LinearSVC` 的效率要高于 `SVC`。

如果你不知道数据集是否为线性，可以直接使用 `SVC` 类创建 SVM 分类器。

在训练和预测中，LinearSVC 和 SVC 一样，都是使用 model.fit(train_X,train_y) 和 model.predict(test_X)。

如何用 SVM 进行乳腺癌检测

在了解了如何创建和使用 SVM 分类器后，我们来看一个实际的项目，数据集来自美国威斯康星州的乳腺癌诊断数据集，[点击这里进行下载](#)。

医疗人员采集了患者乳腺肿块经过细针穿刺 (FNA) 后的数字化图像，并且对这些数字图像进行了特征提取，这些特征可以描述图像中的细胞核呈现。肿瘤可以分成良性和恶性。部分数据截屏如下所示：

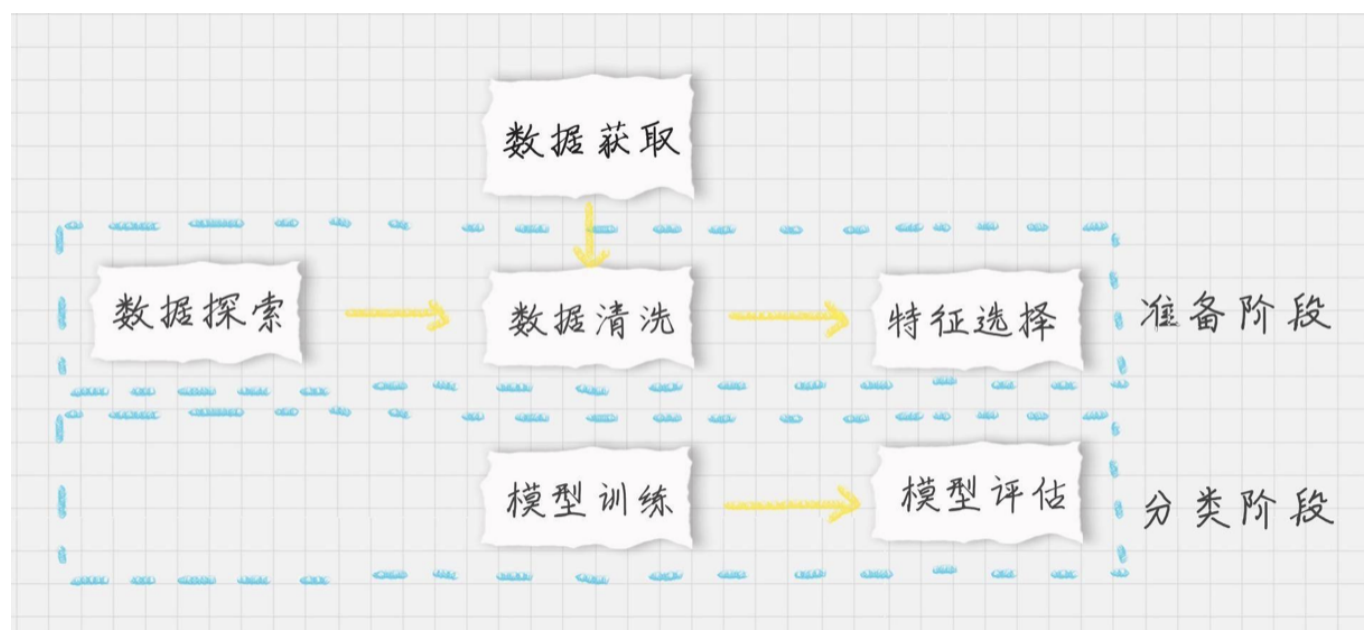
id	diagn	radius	texture	perimeter	area	smoothness	compactness	concavity	concave	symmetry	fractal	radius	texture	perimeter	area	smoothness
842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871	1.095	0.9053	8.589	153.4	0.006399
842517	M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7339	3.398	74.08	0.005225
84300903	M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999	0.7456	0.7869	4.585	94.03	0.00615
84348301	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744	0.4956	1.156	3.445	27.23	0.00911
84358402	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883	0.7572	0.7813	5.438	94.44	0.01149
843786	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613	0.3345	0.8902	2.217	27.19	0.00751
844359	M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.05742	0.4467	0.7732	3.18	53.91	0.004314
84458202	M	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196	0.07451	0.5835	1.377	3.856	50.96	0.008805
844981	M	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389	0.3063	1.002	2.406	24.32	0.005731
84501001	M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.08243	0.2976	1.599	2.039	23.94	0.007149
845636	M	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528	0.05697	0.3795	1.187	2.466	40.51	0.004029
84610002	M	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606	0.1842	0.06082	0.5058	0.9849	3.564	54.16	0.005771
846226	M	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118	0.2397	0.078	0.9555	3.568	11.07	116.2	0.003139
846381	M	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	0.1847	0.05338	0.4033	1.078	2.903	36.58	0.009769
84667401	M	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	0.2069	0.07682	0.2121	1.169	2.061	19.21	0.006429
84799002	M	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303	0.07077	0.37	1.033	2.879	32.55	0.005607
848406	M	14.68	20.13	94.74	684.5	0.09867	0.072	0.07395	0.05259	0.1586	0.05922	0.4727	1.24	3.195	45.4	0.005718
84862001	M	16.13	20.68	108.1	798.8	0.117	0.2022	0.1722	0.1028	0.2164	0.07356	0.5692	1.073	3.854	54.18	0.007025
849014	M	19.81	22.15	130	1260	0.09831	0.1027	0.1479	0.09498	0.1582	0.05395	0.7582	1.017	5.865	112.4	0.006494
8510426	B	13.54	14.36	87.46	566.3	0.09779	0.08129	0.06664	0.04781	0.1885	0.05766	0.2699	0.7886	2.058	23.56	0.008462
8510653	B	13.08	15.71	85.63	520	0.1075	0.127	0.04568	0.0311	0.1967	0.06811	0.1852	0.7477	1.383	14.67	0.004097
8510824	B	9.504	12.44	60.34	273.9	0.1024	0.06492	0.02956	0.02076	0.1815	0.06905	0.2773	0.9768	1.909	15.7	0.009606
8511133	M	15.34	14.26	102.5	704.4	0.1073	0.2135	0.2077	0.09756	0.2521	0.07032	0.4388	0.7096	3.384	44.91	0.006789
851509	M	21.16	23.04	137.2	1404	0.09428	0.1022	0.1097	0.08632	0.1769	0.05278	0.6917	1.127	4.303	93.99	0.004728

数据表一共包括了 32 个字段，代表的含义如下：

字段	含义
ID	ID标识
diagnosis	M/B (M: 恶性, B: 良性)
radius_mean	半径 (点中心到边缘的距离) 平均值
texture_mean	文理 (灰度值的标准差) 平均值
perimeter_mean	周长 平均值
area_mean	面积 平均值
smoothness_mean	平滑程度 (半径内的局部变化) 平均值
compactness_mean	紧密度 ($=\text{周长} \times \text{周长} / \text{面积} - 1.0$) 平均值
concavity_mean	凹度 (轮廓凹部的严重程度) 平均值
concave points_mean	凹缝 (轮廓的凹部分) 平均值
symmetry_mean	对称性 平均值
fractal_dimension_mean	分形维数 ($=\text{海岸线近似} - 1$) 平均值
radius_se	半径 (点中心到边缘的距离) 标准差
texture_se	文理 (灰度值的标准差) 标准差
perimeter_se	周长 标准差
area_se	面积 标准差
smoothness_se	平滑程度 (半径内的局部变化) 标准差
compactness_se	紧密度 ($=\text{周长} \times \text{周长} / \text{面积} - 1.0$) 标准差
concavity_se	凹度 (轮廓凹部的严重程度) 标准差
concave points_se	凹缝 (轮廓的凹部分) 标准差
symmetry_se	对称性标准差
fractal_dimension_se	分形维数 ($=\text{海岸线近似} - 1$) 标准差
radius_worst	半径 (点中心到边缘的距离) 最大值
texture_worst	文理 (灰度值的标准差) 最大值
perimeter_worst	周长 最大值
area_worst	面积 最大值
smoothness_worst	平滑程度 (半径内的局部变化) 最大值
compactness_worst	紧密度 ($=\text{周长} \times \text{周长} / \text{面积} - 1.0$) 最大值
concavity_worst	凹度 (轮廓凹部的严重程度) 最大值
concave points_worst	凹缝 (轮廓的凹部分) 最大值
symmetry_worst	对称性 最大值
fractal_dimension_worst	分形维数 ($=\text{海岸线近似} - 1$) 最大值


上面的表格中，mean 代表平均值，se 代表标准差，worst 代表最大值（3 个最大值的平均值）。每张图像都计算了相应的特征，得出了这 30 个特征值（不包括 ID 字段和分类标识结果字段 diagnosis），实际上是 10 个特征值（radius、texture、perimeter、area、smoothness、compactness、concavity、concave points、symmetry 和 fractal_dimension_mean）的 3 个维度，平均、标准差和最大值。这些特征值都保留了 4 位数字。字段中没有缺失的值。在 569 个患者中，一共有 357 个是良性，212 个是恶性。

好了，我们的目标是生成一个乳腺癌诊断的 SVM 分类器，并计算这个分类器的准确率。首先设定项目的执行流程：



1. 首先我们需要加载数据源；
2. 在准备阶段，需要对加载的数据源进行探索，查看样本特征和特征值，这个过程你也可以使用数据可视化，它可以方便我们对数据及数据之间的关系进一步加深了解。然后按照“完全合一”的准则来评估数据的质量，如果数据质量不高就需要做数据清洗。数据清洗之后，你可以做特征选择，方便后续的模型训练；
3. 在分类阶段，选择核函数进行训练，如果不知道数据是否为线性，可以考虑使用 `SVC(kernel= 'rbf')`，也就是高斯核函数的 SVM 分类器。然后对训练好的模型用测试集进行评估。

按照上面的流程，我们来编写下代码，加载数据并对数据做部分的探索：

 复制代码


```
1 # 加载数据集，你需要把数据放到目录中
2 data = pd.read_csv("./data.csv")
```

```

3 # 数据探索
4 # 因为数据集中列比较多，我们需要把 dataframe 中的列全部显示出来
5 pd.set_option('display.max_columns', None)
6 print(data.columns)
7 print(data.head(5))
8 print(data.describe())

```

这是部分的运行结果，完整结果你可以自己跑一下。

 复制代码


```

1 Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
2       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
3       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
4       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
5       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
6       'fractal_dimension_se', 'radius_worst', 'texture_worst',
7       'perimeter_worst', 'area_worst', 'smoothness_worst',
8       'compactness_worst', 'concavity_worst', 'concave points_worst',
9       'symmetry_worst', 'fractal_dimension_worst'],
10      dtype='object')
11      id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean \
12  0      842302         M      17.99      10.38      122.80      1001.0
13  1      842517         M      20.57      17.77      132.90      1326.0
14  2      84300903        M      19.69      21.25      130.00      1203.0
15  3      84348301         M      11.42      20.38       77.58       386.1
16  4      84358402         M      20.29      14.34      135.10      1297.0

```

接下来，我们就要对数据进行清洗了。

运行结果中，你能看到 32 个字段里，id 是没有实际含义的，可以去掉。diagnosis 字段的取值为 B 或者 M，我们可以用 0 和 1 来替代。另外其余的 30 个字段，其实可以分成三组字段，下划线后面的 mean、se 和 worst 代表了每组字段不同的度量方式，分别是平均值、标准差和最大值。

 复制代码


```

1 # 将特征字段分成 3 组
2 features_mean= list(data.columns[2:12])
3 features_se= list(data.columns[12:22])
4 features_worst=list(data.columns[22:32])
5 # 数据清洗

```

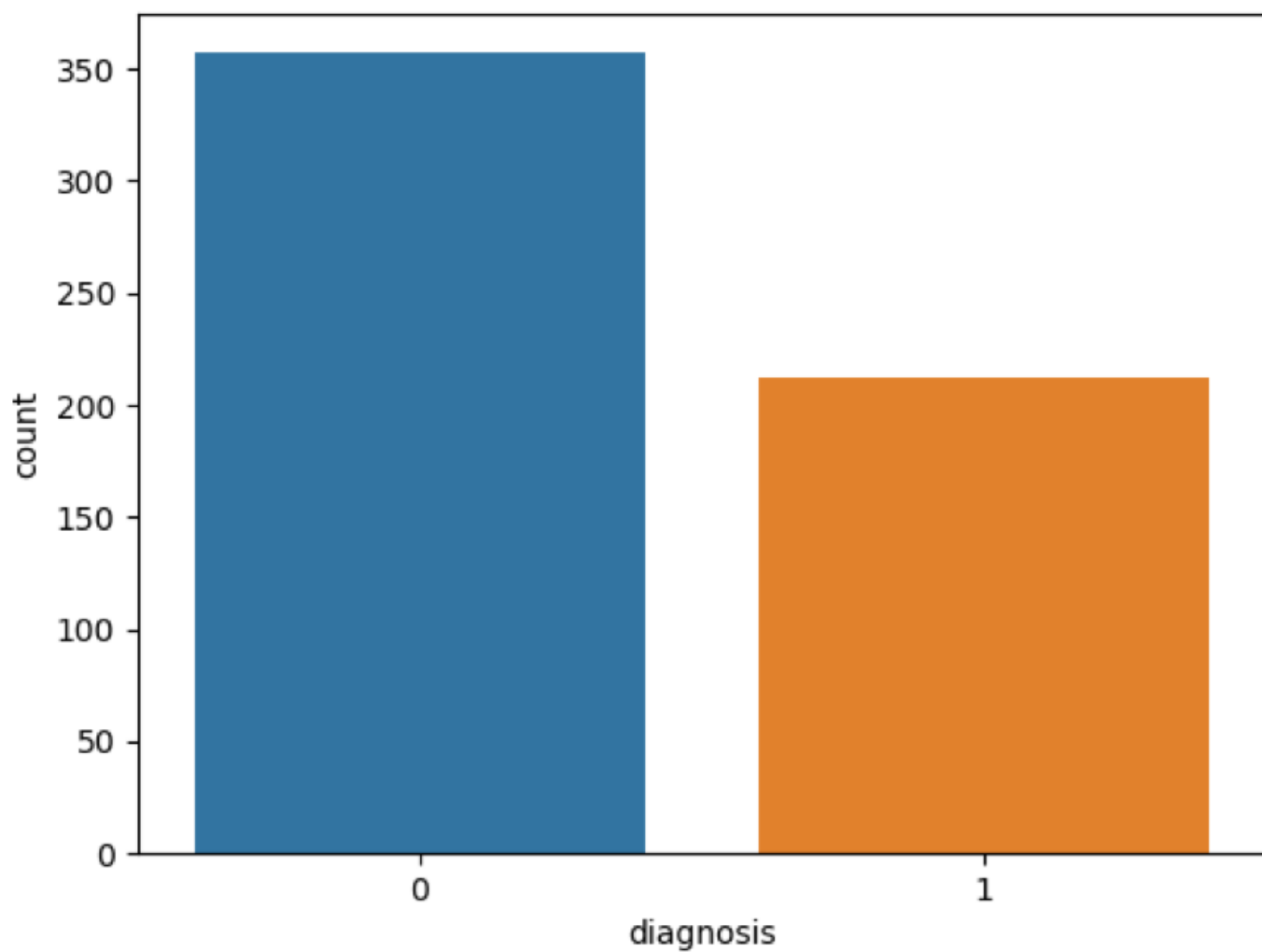
```
6 # ID 列没有用，删除该列
7 data.drop("id",axis=1,inplace=True)
8 # 将 B 良性替换为 0，M 恶性替换为 1
9 data['diagnosis']=data['diagnosis'].map({'M':1,'B':0})
```

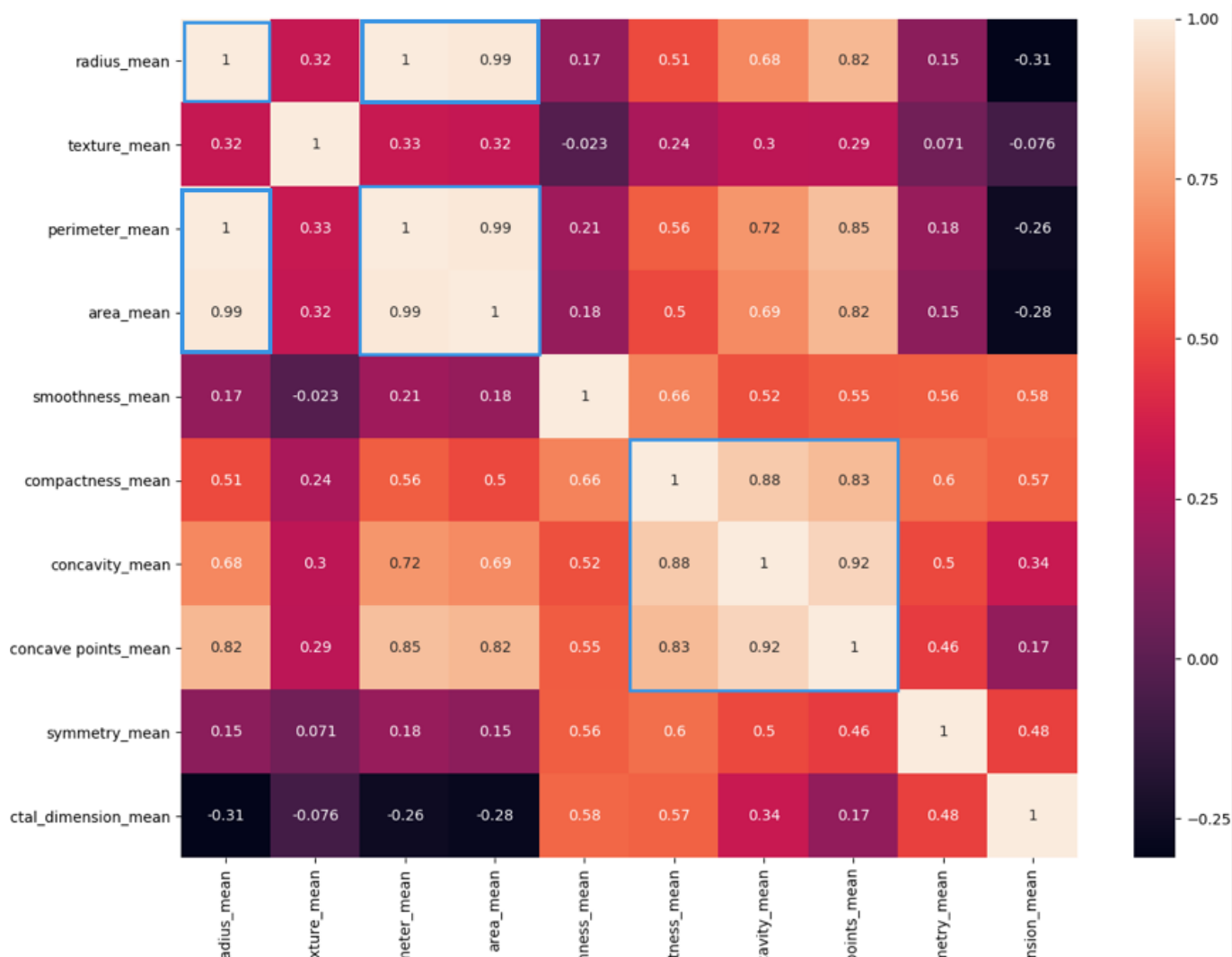
然后我们要做特征字段的筛选，首先需要观察下 features_mean 各变量之间的关系，这里我们可以用 DataFrame 的 corr() 函数，然后用热力图帮我们可视化呈现。同样，我们也会看整体良性、恶性肿瘤的诊断情况。

 复制代码

```
1 # 将肿瘤诊断结果可视化
2 sns.countplot(data['diagnosis'],label="Count")
3 plt.show()
4 # 用热力图呈现 features_mean 字段之间的相关性
5 corr = data[features_mean].corr()
6 plt.figure(figsize=(14,14))
7 # annot=True 显示每个方格的数据
8 sns.heatmap(corr, annot=True)
9 plt.show()
```

这是运行的结果：






热力图中对角线上的为单变量自身的相关系数是 1。颜色越浅代表相关性越大。所以你能看出来 radius_mean、perimeter_mean 和 area_mean 相关性非常大，compactness_mean、concavity_mean、concave points_mean 这三个字段也是相关的，因此我们可以取其中的一个作为代表。

那么如何进行特征选择呢？

特征选择的目的是降维，用少量的特征代表数据的特性，这样也可以增强分类器的泛化能力，避免数据过拟合。


我们能看到 mean、se 和 worst 这三组特征是对同一组内容的不同度量方式，我们可以保留 mean 这组特征，在特征选择中忽略掉 se 和 worst。同时我们能看到 mean 这组特征中，radius_mean、perimeter_mean、area_mean 这三个属性相关性大，compactness_mean、daconcavity_mean、concave points_mean 这三个属性相关性大。我们分别从这 2 类中选择 1 个属性作为代表，比如 radius_mean 和 compactness_mean。

这样我们就可以把原来的 10 个属性缩减为 6 个属性，代码如下：

 复制代码

```
1 # 特征选择
2 features_remain = ['radius_mean', 'texture_mean', 'smoothness_mean', 'compactness_mean', '':
```

对特征进行选择之后，我们就可以准备训练集和测试集：

 复制代码

```
1 # 抽取 30% 的数据作为测试集，其余作为训练集
2 train, test = train_test_split(data, test_size = 0.3)# in this our main data is splitted
3 # 抽取特征选择的数值作为训练和测试数据
4 train_X = train[features_remain]
5 train_y=train['diagnosis']
6 test_X= test[features_remain]
7 test_y =test['diagnosis']
```

在训练之前，我们需要对数据进行规范化，这样让数据同在同一个量级上，避免因为维度问题造成数据误差：

 复制代码

```
1 # 采用 Z-Score 规范化数据，保证每个特征维度的数据均值为 0，方差为 1
2 ss = StandardScaler()
3 train_X = ss.fit_transform(train_X)
4 test_X = ss.transform(test_X)
```

最后我们可以让 SVM 做训练和预测了：

 复制代码

```
1 # 创建 SVM 分类器
2 model = svm.SVC()
3 # 用训练集做训练
4 model.fit(train_X, train_y)
5 # 用测试集做预测
6 prediction=model.predict(test_X)
7 print('准确率：', metrics.accuracy_score(prediction, test_y))
```

运行结果：

复制代码

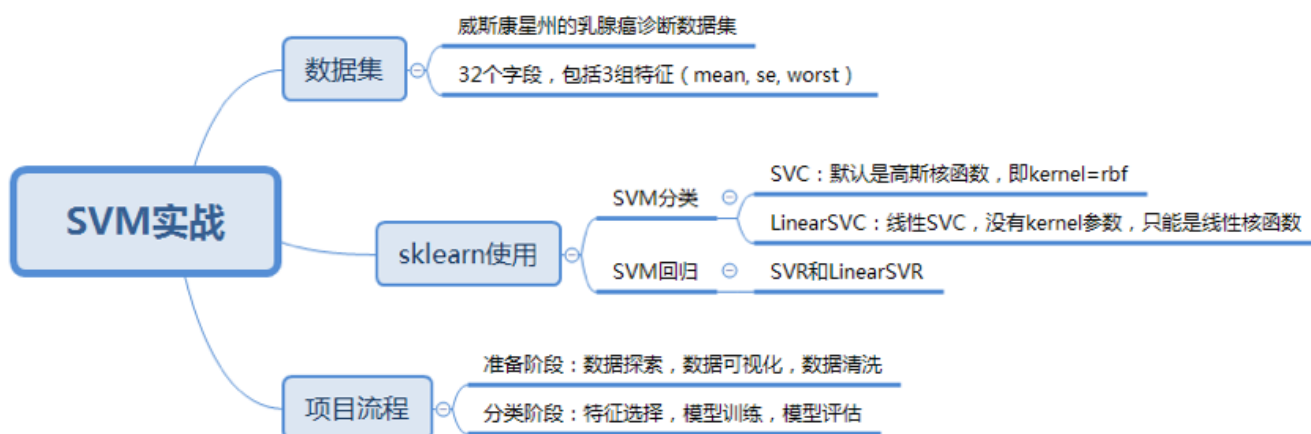
```
1 准确率： 0.9181286549707602
```

准确率大于 90%，说明训练结果还不错。完整的代码你可以从[GitHub](#)上下载。

总结

今天我带你一起做了乳腺癌诊断分类的 SVM 实战，从这个过程中你应该能体会出来整个执行的流程，包括数据加载、数据探索、数据清洗、特征选择、SVM 训练和结果评估等环节。

sklearn 已经为我们提供了很好的工具，对上节课中讲到的 SVM 的创建和训练都进行了封装，让我们无需关心中间的运算细节。但正因为这样，我们更需要对每个流程熟练掌握，通过实战项目训练数据化思维和对数据的敏感度。



最后给你留两道思考题吧。还是这个乳腺癌诊断的数据，请你用 LinearSVC，选取全部的特征（除了 ID 以外）作为训练数据，看下你的分类器能得到多少的准确度呢？另外你对 sklearn 中 SVM 使用又有怎样的体会呢？

欢迎在评论区与我分享你的答案，也欢迎点击“请朋友读”，把这篇文章分享给你的朋友或者同事，一起来交流，一起来进步。

数据分析实战 45 讲

即学即用的数据分析入门课

陈旻

清华大学计算机博士



新版升级：点击「👤 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 22 | SVM（上）：如何用一根棍子将蓝红两色球分开？

下一篇 24 | KNN（上）：如何根据打斗和接吻次数来划分电影类型？

精选留言 (20)

写留言



reverse

2019-03-06

6

极客时间数据分析实战45讲的详细笔记(包含markdown、图片、思维导图 代码) github地址：<https://github.com/xiaomiwujiecao/DataAnalysisInAction>

展开



滢

2019-04-18

3

利用SVM做分类，特征选择影响度大，要想SVM分类准确，人工处理数据这一步很重要



Geek_dance...

2019-02-27



默认SVC训练模型，6个特征变量，训练集准确率：96.0%，测试集准确率：92.4%

默认SVC训练模型，10个特征变量，训练集准确率：98.7%，测试集准确率：98.2%

LinearSVC训练模型，6个特征变量，训练集准确率：93.9%，测试集准确率：92.3%

LinearSVC训练模型，10个特征变量，训练集准确率：99.4%，测试集准确率：96.0%

...

展开 ∨



mickey

2019-02-26



```
# encoding=utf-8
```

```
from sklearn import svm
```

```
from sklearn import metrics
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import StandardScaler...
```

展开 ∨



Rickie

2019-02-05



思考题：

使用全部数据进行训练得到的准确率为0.9766，高于示例中的准确率。是否是由于多重共线性，使得测试结果偏高？

展开 ∨



hlz-123

2019-03-16



首先要说，老师的课讲得非常好，深奥的算法和理论通过生动有趣的例子让人通俗易懂，兴趣盎然。

老师的本课案例中，对特征数据都做了Z-Score规范化处理（正态分布），准确率在90%以上，如果数据不做规范化处理，准确率在88%左右，我的问题：

1、数据规范化处理，是不是人为地提供了准确率？实际情况，数据不一定是正态分布。...

展开 ∨



third

2019-02-18



第二个，准确率 0.935672514619883。

感觉还蛮好用的，只是不是很熟练的使用各个算法做分类和回归

展开 ▾



张晓辉

2019-05-19



采用linearSVC, 预测准确率更高。

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score...
```

展开 ▾



滢

2019-04-18



语言Python3.6 没有z-score规范化数据以及规范化后两种情况前提预测准确率，使用LinearSVC，选取所有mean属性

```
import pandas as pd
import matplotlib.pyplot as plt...
```

展开 ▾



王彬成

2019-04-02



```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler...
```

展开 ▾



圆圆的大食...

2019-03-18



```
# -*- coding: utf-8 -*-
"""
```

Created on Sun Mar 17 23:18:31 2019

@author: xcma1...

展开 ▾



叮当猫

2019-03-16



通过多次运行，发现每次结果也不太一样，但是大体上，svc的效果好于linearsvc，多特征好于少特征。

附代码如下：

```
#-coding=utf-8...
```

展开 ▾



三硝基甲苯

2019-03-10



用K折交叉验证，LinearSVC的准确率是92.64% SVC是92.98%

至于SVC的使用，我一开始直接按照自己的想法写完以后，会有聚合警告，然后看了一下，是数据没有进行StandardScaler，我觉得这个步骤容易忘记。

展开 ▾



fancy

2019-03-02



使用LinearSVC和全部特征作为训练集时，分类器的准确率达到99.4152%，在其他条件不变的情况下，其准确率高于SVC。



ldw

2019-02-28



陈老师，这节课留的课后任务，包括可能使用的数据清洗，您会期望您团队的人用多长时间完成？超过多长时间以上，就是不合格的？谢谢 ☺



mickey

2019-02-26



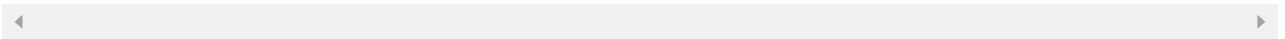
勘误：热力学图中的第一个蓝色框框应该是标记在第1列第3-4行上，而不是第1列第1行。

编辑回复: 代表的含义是: radius_mean, perimeter_mean, area_mean这三个指标正相关, 因此选择其中一个代表即可 (我在正文中也写到了)

你说的标注第一列第3-4行也是对的, 因为这几个指标都是正相关。完整的看第一行的第3-4列也可以标注上, 实际上这三个指标可以重新组成一个小矩形。

我的标注 (第一行第一列+第34行第34列, 代表的是这三个指标相关) 起到提示的作用, 最主要的还是说明: radius_mean, perimeter_mean, area_mean这三个指标正相关。这个是最终的结果。

Anyway 你把第一列第3-4行标注出来, 或者第一行第3-4列标注出来都是对的



Destroy、

2019-02-20



特征选择

```
features_all = ['radius_mean', 'texture_mean', 'perimeter_mean',  
               'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',  
               'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',  
               'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',...
```

展开 ∨



深白浅黑

2019-02-15



使用全部特征: (相同训练集和测试集)

LinearSVC准确率: 0.9298245614035088

SVC高斯核准确率: 0.9415204678362573

SVM首先是有监督的学习模型, 需要数据有较好的分类属性。其次依据硬间隔、软间隔和核函数的应用, 可以解决线性分类和非线性分类的问题。最后在使用过程中, 需要对数...

展开 ∨



JingZ

2019-02-15



#svm 使用还是蛮方便的, 完全特征, 准确率达到97%以上

```
import pandas as pd  
from sklearn.model_selection import train_test_split  
from sklearn.preprocessing import StandardScaler...
```

展开 ∨



Python

2019-02-04



老师可以用PCA进行特征选择吗？如果可以，那和你这种手动的方法比有什么差别