Lab 5 Report

戴维 PB22151783

1. 任务描述

在实验3，4中我们对两个乳腺癌数据集进行了数据分析。在实验5中我们希望通过数据集中的数据对两个关键特征 `diagnosis` 和 `Class` 进行分类预测。

2. 实验3 diagnosis 预测

2.1 数据信息和预处理

实验3中我们使用了威斯康辛州乳腺癌数据集（Breast Cancer Wisconsin Dataset），有31个特征（除患者ID，ID对分类没有帮助，因此我们将ID删去），569条数据。其中smoothness_mean, fractal_dimension_mean, texture_se, compactness_se, concavity_se, fractal_dimension_se, radius_worst, smoothness_worst 共8个特征存在含有缺失值的记录，有缺失值的记录共9条。由于含有缺失值的记录约占数据集总数的1.58%，去除后对数据集整体影响不大，因此我们在此将含有缺失值的记录全部删去。此外，关于我们关心的 diagnosis，560条数据记录中有355条记录为 B，代表肿瘤为良性；205条记录为M，代表肿瘤为恶性。由此可以得出数据集关于特征 diagnosis 分布较为均匀。

2.2 数据集划分

我们按 8 : 2 的比例将数据集划分为训练集和测试集。为了在分类模型上获得表现较优的超参数，我们在训练集上使用 k 折交叉验证的方法，每次从中取出部分数据作为验证集，综合 k 折交叉验证的结果，由此我们可以得出表现较好的超参数。

2.3 分类算法模型

本次实验中，我们选用随机森林和支持向量机两个分类算法模型进行分类预测。

2.3.1 随机森林

随机森林是一种集成学习方法，通过在训练数据集中随机抽取多个子样本来构建多棵决策树，每棵决策树独立生成并进行分类，最终的分类结果通过对所有树的投票结果来决定。随机森林利用了Bagging（自助法）和随机特征选择技术，能够有效减少过拟合，提升模型的泛化能力。随机森林模型的优缺点如下：

1. **优点**：
   - **鲁棒性强**：能够处理高维数据且不容易过拟合。

- **处理缺失值**：对缺失值有较好的处理能力。
- **特征重要性**：可以评估特征的重要性，提供额外的信息。

2. **缺点**：

- **计算开销**：由于需要构建大量的决策树，计算和内存开销较大。
- **模型复杂度**：难以解释单个树的决策过程，整体模型较为复杂。

本次实验中我们选择直接调用算法库 `sklearn.ensemble` 中的类 `RandomForestClassifier` 来实现模型预测。类 `RandomForestClassifier` 具有 `.fit()`， `.predict()` 等方法用于训练和预测；并有以下可调整的超参数：

超参数	类型	默认值	描述
`n_estimators`	`int`	`100`	森林中树的数量。
`criterion`	`{"gini", "entropy", "log_loss"}`	`"gini"`	用于衡量拆分质量的函数。
`max_depth`	`int`, `None`	`None`	树的最大深度。 `None` 表示树生长直到所有叶子节点纯净或达到最小样本数为止。
`min_samples_split`	`int`, `float`	`2`	分裂一个内部节点所需的最小样本数。
`min_samples_leaf`	`int`, `float`	`1`	叶子节点所需的最小样本数。
`min_weight_fraction_leaf`	`float`	`0.0`	叶子节点中样本权重的最小加权分数。
`max_features`	`{"auto", "sqrt", "log2"}`, `int`, `float`, `None`	`None`	寻找最佳分裂时要考虑的特征数量。
`max_leaf_nodes`	`int`, `None`	`None`	叶子节点的最大数量。 `None` 表示不限制叶子节点的数量。
`min_impurity_decrease`	`float`	`0.0`	分裂节点时需要的最小不纯度减少量。
`bootstrap`	`bool`	`True`	是否在构建树时使用自助法（bootstrap）抽样。
`oob_score`	`bool`	`False`	是否使用袋外样本来估计泛化精度。
`n_jobs`	`int`, `None`	`None`	并行运行时使用的CPU核数。 `None` 表示 `1`， `-1` 表示使用所有处理器。

超参数	类型	默认值	描述
`random_state`	`int`, `RandomState`, `None`	`None`	控制随机性。
`verbose`	`int`	`0`	控制树构建时的详细程度。
`warm_start`	`bool`	`False`	是否使用前一次调用的解决方案初始化，以增加更多的树。
`class_weight`	`dict`, `list of dict`, `str`, `None`	`None`	与类关联的权重。`balanced` 模式自动调整权重以处理不平衡数据。
`ccp_alpha`	`float`	`0.0`	最小成本复杂度修剪参数。
`max_samples`	`int`, `float`, `None`	`None`	从每棵树的引导样本中抽取的最大样本数。如果是浮点数，则表示样本数量的比例。

- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32.
- Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news*, 2(3), 18-22.
- RandomForestClassifier Documentation

2.3.2 支持向量机

支持向量机是一种基于统计学习理论的监督学习算法，广泛用于分类和回归分析。SVM通过在高维空间中寻找一个最优的分离超平面，使得不同类别的数据点间的间隔最大化。对于线性不可分的数据，SVM使用核函数将数据映射到高维空间，在高维空间中实现线性分离。支持向量机的优缺点如下：

1. **优点**：
   - **高效**：在高维空间中依然表现良好，尤其适合处理维数较高的数据。
   - **灵活性**：通过选择不同的核函数（如线性核、多项式核、径向基核等）能够适应不同的数据分布。
   - **防止过拟合**：通过正则化参数控制模型复杂度，有效防止过拟合。
2. **缺点**：
   - **计算复杂度**：对大规模数据集，训练过程计算量大，速度较慢。
   - **参数选择**：需要对核函数和参数进行选择和调整，模型调参较为复杂。

本次实验我们通过调用 `sklearn.svm` 中的 `SVC` 类实现分类预测， `SVC` 的基本方法与上文随机森林类似，其具有以下可以调节的超参数：

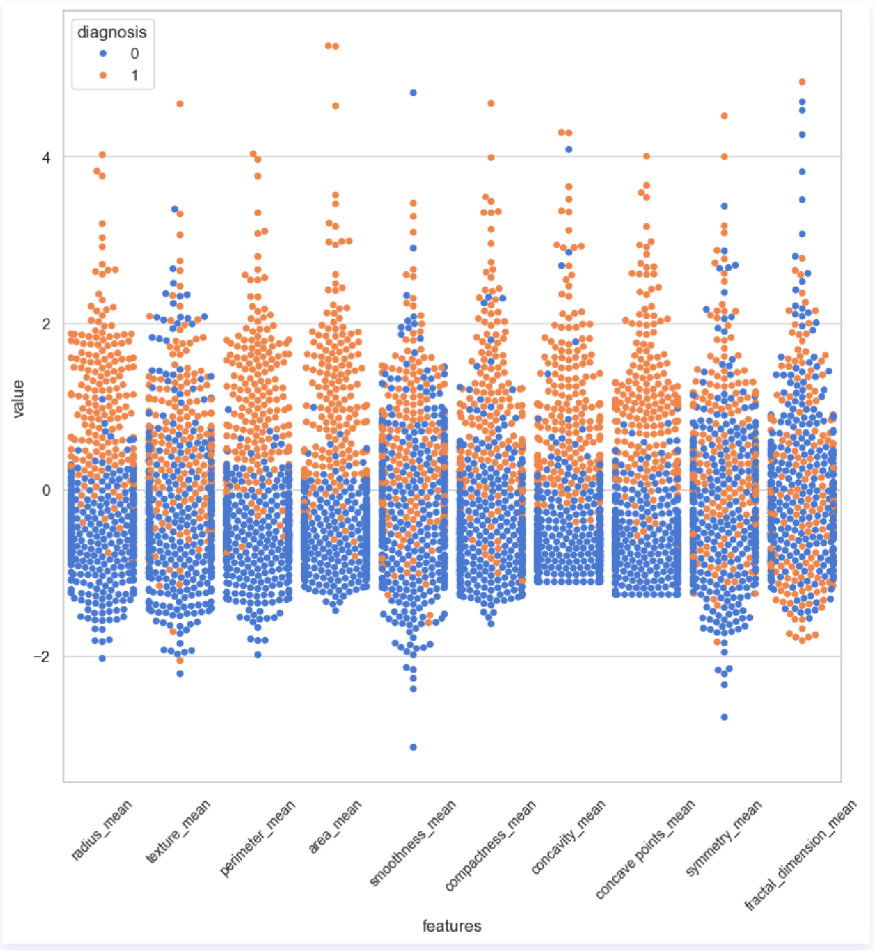
超参数	类型	默认值	描述
`C`	`float`	`1.0`	正则化参数。必须是正浮点数。较小的值表示更强的正则化。
`kernel`	`{"linear", "poly", "rbf", "sigmoid", "precomputed"}`	`"rbf"`	指定用于决策函数的内核类型。
`degree`	`int`	`3`	多项式核函数的维度。仅在核函数为 `poly` 时有效。
`gamma`	`{"scale", "auto"}`, `float`	`"scale"`	核函数系数。
`coef0`	`float`	`0.0`	核函数中的独立项。它只在 `poly` 和 `sigmoid` 核函数中有效。
`shrinking`	`bool`	`True`	是否使用启发式方法加速训练过程。
`probability`	`bool`	`False`	是否启用概率估计。启用后会显著增加训练时间。
`tol`	`float`	`1e-3`	停止训练的误差容忍度。
`cache_size`	`float`	`200`	指定内核缓存的大小（以MB为单位）。
`class_weight`	`dict`, `{"balanced"}`, `None`	`None`	给定各类的权重。 `balanced` 模式根据类频率自动调整权重以处理不平衡数据。
`verbose`	`bool`	`False`	启用详细输出。
`max_iter`	`int`	`-1`	最大迭代次数。 `-1` 表示无限制。
`decision_function_shape`	`{"ovo", "ovr"}`	`"ovr"`	指定决策函数的形状。 `ovr` 为一对多策略， `ovo` 为一对一策略。
`break_ties`	`bool`	`False`	如果为 `True` ，则 `predict` 方法将打破平局（仅适用于 `ovr` ）。

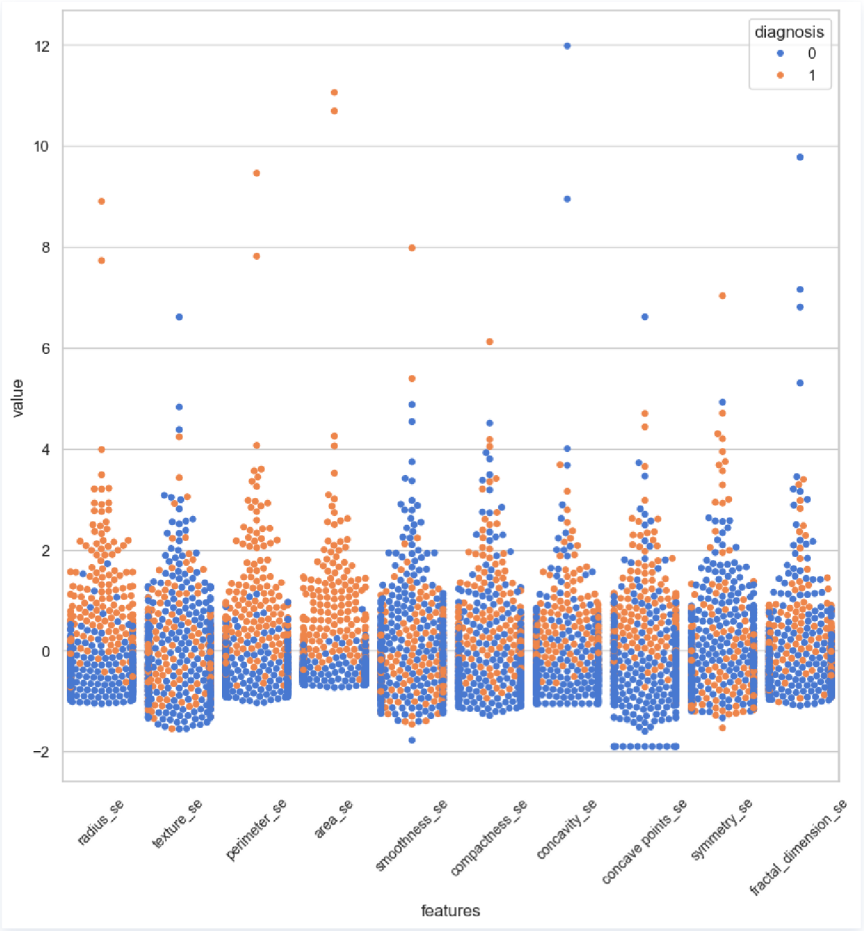
超参数	类型	默认值	描述
`random_state`	`int`, `RandomState`, `None`	`None`	控制随机数生成器的种子，用于概率估计数据重排和 `break_ties=True` 时平局打破。

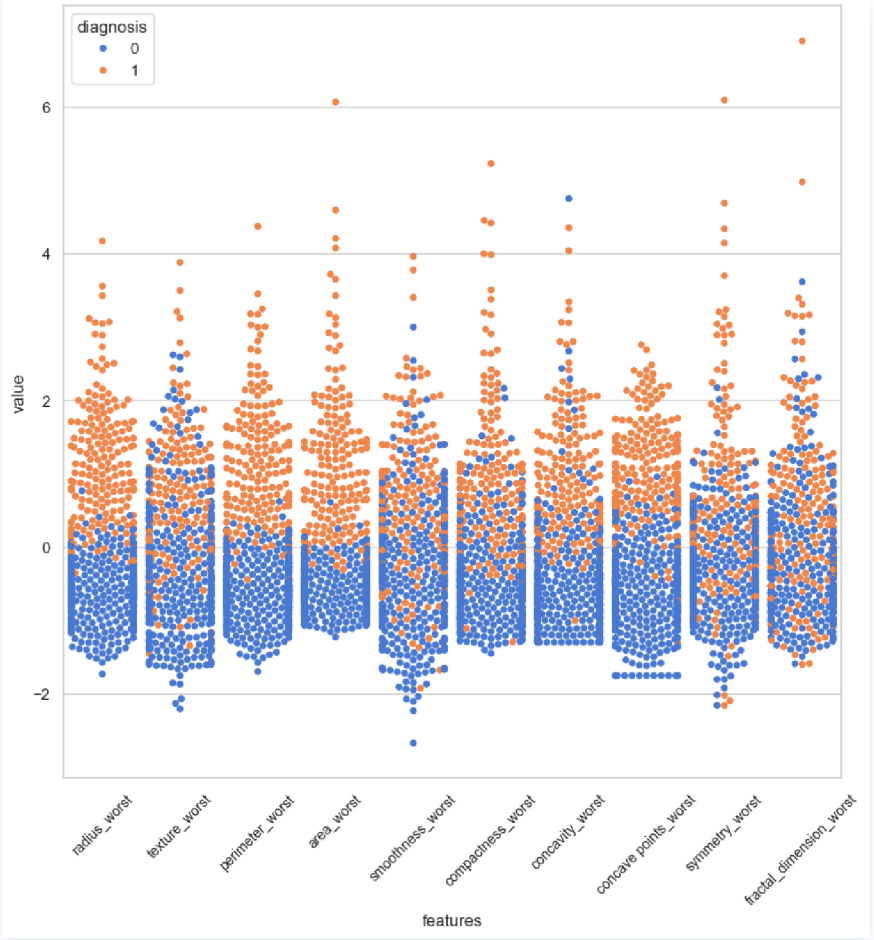
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273-297.
- Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2, 121-167.
- SVC Documentation

2.4 特征选择与处理

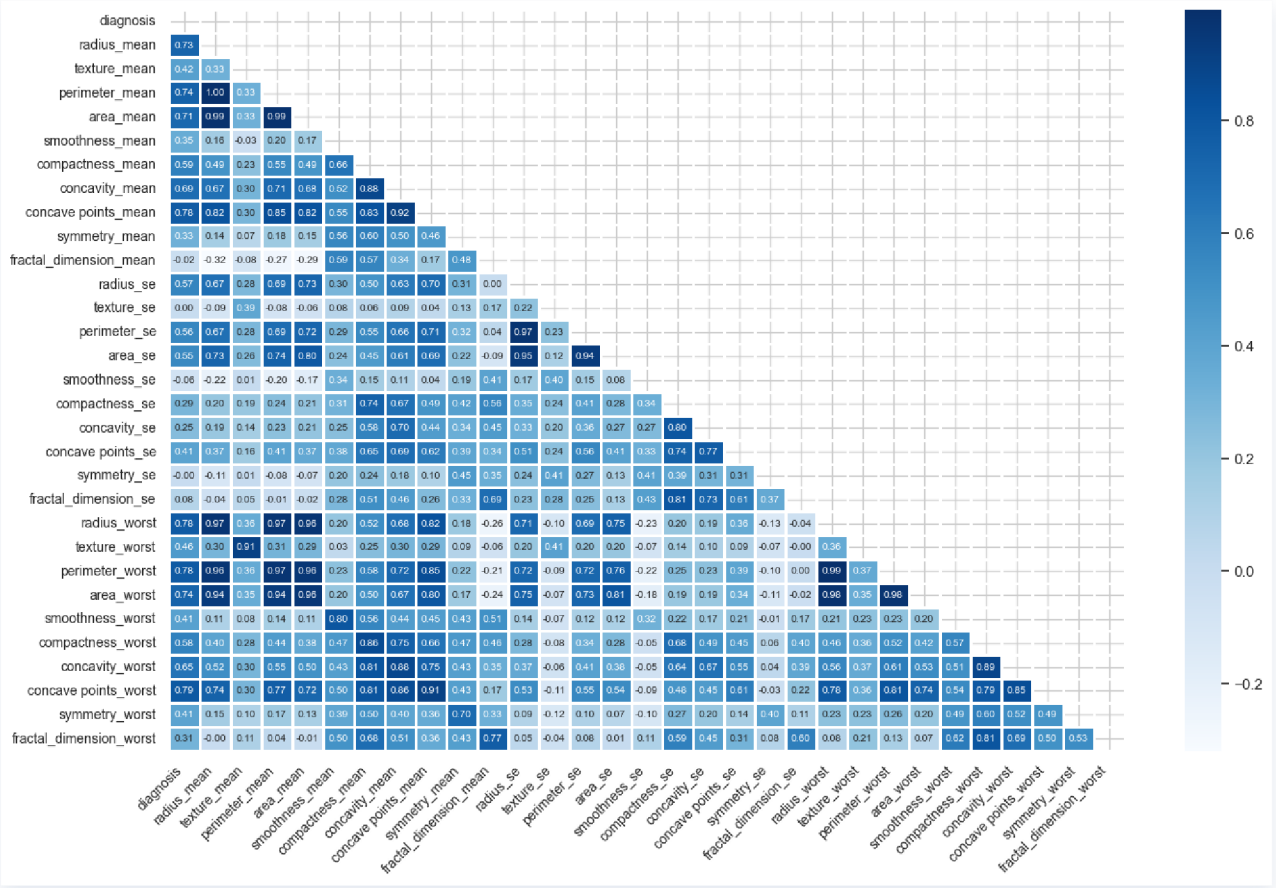
为了选取出有价值的特征，我们先尝试从图像入手。由于数据集中的特征均为连续值，为了便于数据可视化，我们首先将数据标准化。我们将标准化后的数据绘制成分类散点图，结果如下图所示：




diagnosis
0
1
value
features
radius_se
texture_se
perimeter_se
area_se
smoothness_se
compactness_se
concavity_se
concave points_se
symmetry_se
fractal_dimension_se


diagnosis
0
1
value
features
radius_worst
texture_worst
perimeter_worst
area_worst
smoothness_worst
compactness_worst
concavity_worst
concave points_worst
symmetry_worst
fractal_dimension_worst

由图像我们可以得出， radius_worst, perimeter_worst, area_worst 等单一特征可以较好得完成分类。但为了提高分类的准确率和泛化性，考虑到其余特征能够组成复合的判断指标完成分类预测，我们进一步研究数据的关联矩阵。数据的关联矩阵如下图所示：



结合关联矩阵的结果以及特征的实际含义，我们可以得知 radius_mean, area_mean, perimeter_mean 等特征高度关联。由于本数据集特征较多，我们可以考虑去除一些冗余的特征，在加速分类的同时，一定程度上也可以提高模型的泛化能力。观察关联矩阵，取关联矩阵的下三角矩阵，选择关联值的绝对值大于0.92的特征（由于我们取下三角子矩阵，避免了重复去除元素）。高度相关的特征有 'radius_mean', 'perimeter_mean', 'area_mean', 'radius_se', 'perimeter_se', 'radius_worst', 'perimeter_worst'，在此我们只保留 'radius_mean' 这一个特征。

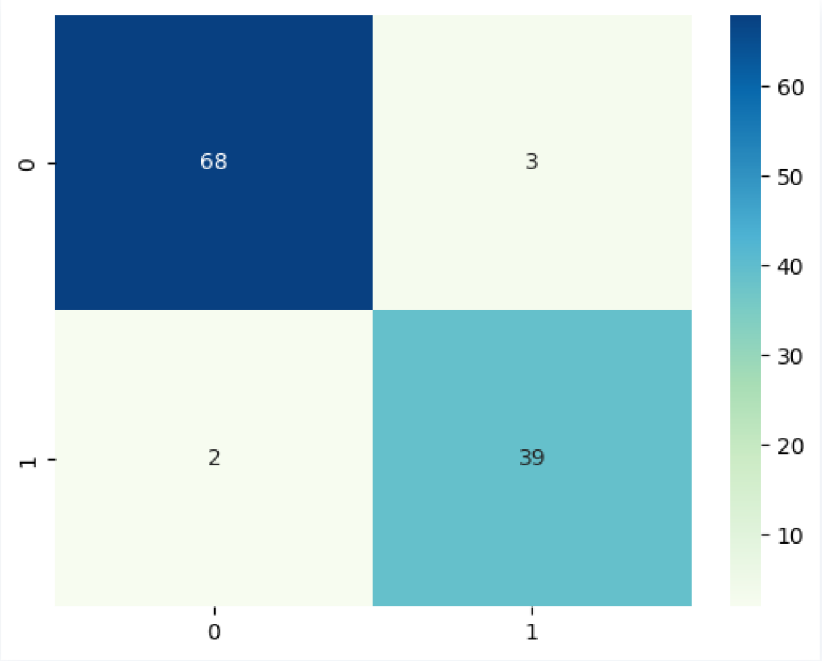
2.5 主试验

2.5.1 随机森林

我们调用 `sklearn` 中的 `RandomForestClassifier` 类，设置 `random_state=0` 便于结果复现，其余超参数均保持默认值。我们在训练集和测试集上 Accuracy 分别达到了 1.0，0.955。其他指标与混淆矩阵如下所示：

指标	Diagnosis 0	Diagnosis 1
Precision	0.97	0.93
Recall	0.96	0.95

指标	Diagnosis 0	Diagnosis 1
F1-Score	0.96	0.94
Support	71	41

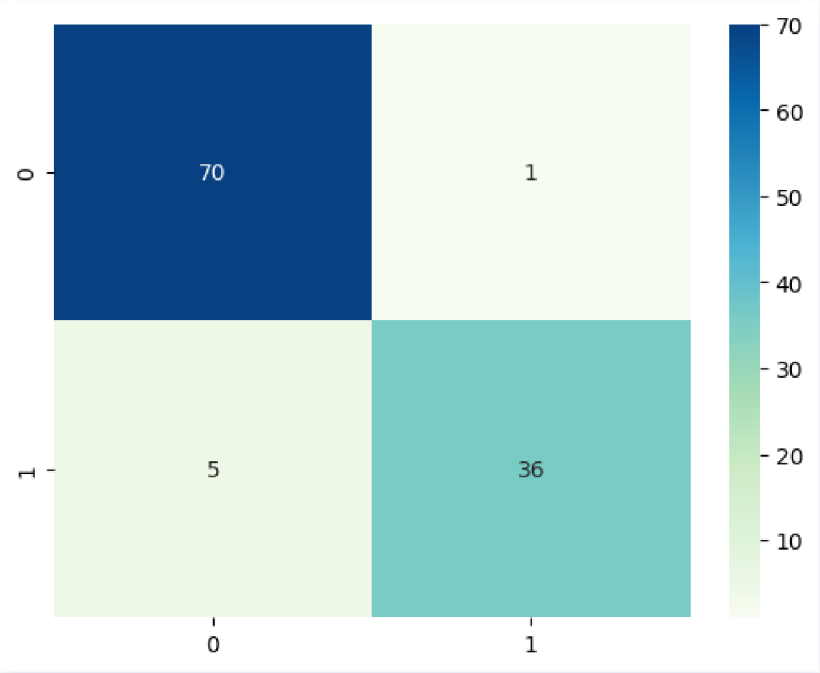


由上面的指标我们可以发现我们的模型可以很好地完成对 `diagnosis` 的预测，具有较高的正确率，并且具有较好的泛化性。

2.5.2 支持向量机

同样的，我们调用 `sklearn` 中的 `SVC` 类来实现分类预测，设置 `random_state=0, probability=True`，其余超参数保持默认，这里我们设置 `probability=True` 以便能够计算预测概率，方便后续 ROC 曲线和 AUC 的计算。我们在训练集和测试集上 Accuracy 分别达到了 0.915，0.946。其他指标与混淆矩阵如下所示：

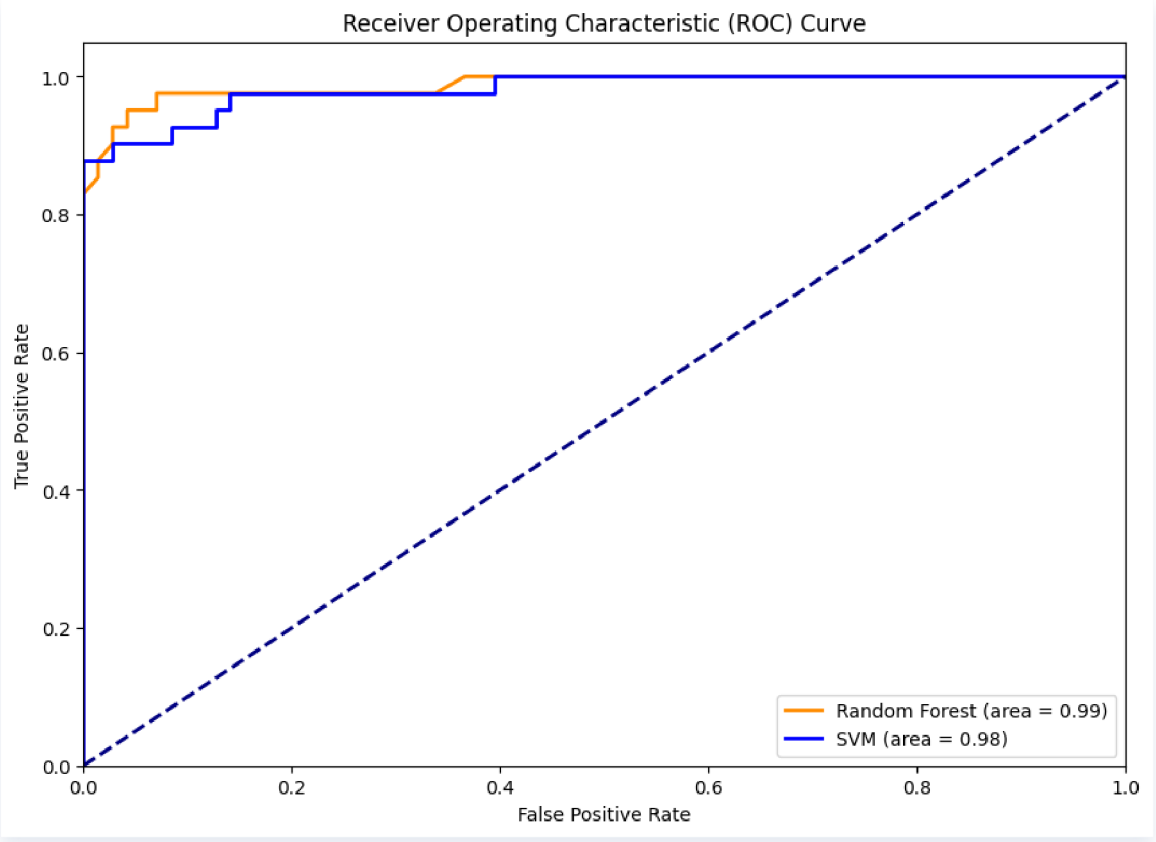
指标	Diagnosis 0	Diagnosis 1
Precision	0.93	0.97
Recall	0.99	0.88
F1-Score	0.96	0.92
Support	71	41



由上述指标我们可以得出：支持向量机也很好的实现了对 `diagnosis` 的预测，具有较高的正确率。

2.5.3 模型对比

ROC-AUC能够综合地反应模型的性能，在此我们绘制出上述两个模型的ROC曲线图如下：



从图中我们可以发现，随机森林和支持向量机均有较强的泛化能力，由AUC我们可以得出：在该分类任务上，随机森林的综合性能要略优于支持向量机。

2.6 参数实验

`sklearn` 中提供了 `GridSearchCV` 类帮助我们获得较优的模型超参数。这里我们通过设置 `GridSearchCV` 的属性 `cv=5` 来实现 5 折交叉验证获取验证集，检测超参数表现的效果。

2.6.1 随机森林参数调整

我们为网格搜索设置如下所示的超参数列表：

参数	取值
n_estimators	20, 50, 100
max_depth	None, 5, 10
min_samples_split	2, 5, 10
min_samples_leaf	1, 2, 4
max_features	auto, sqrt, log2

通过网格搜索我们确定了一组较优的超参数为'max_depth': 5, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 100。我们以上述超参数建立 `RandomForestClassifier`，保持 `random_state=0`，在新的超参数下，模型在测试集上的准确率提高到了0.9821428571428571。

2.6.2 支持向量机参数调整

我们为支持向量机的网格搜索设置如下的超参数列表：

参数	取值范围
C	0.1, 1, 10, 100
gamma	0.01, 0.1, 1, 10
kernel	linear, rbf

通过计算我们得出该任务下支持向量机的一组较优超参数为 'C': 10, 'gamma': 0.01, 'kernel': 'linear'。使用上述超参数，模型在测试集上的准确率提高到了0.9910714285714286，效果十分显著。

3. 实验4 Class 预测

3.1 数据信息和预处理

实验4中我们使用了来自1988年南斯拉夫的乳腺癌数据集（Breast Cancer Dataset）。数据集共有286条数据，10个特征（除id外，id不属于特征，故我们将id特征删去）。特征 `node-caps` 和 `breast-quad` 存在缺失值，共有9条数据存在缺失值，由于含缺失值的数据占比约为 3.15%，我们选择忽略含有缺失值的记录。我们关注的预测特征为 `Class`，该特征代表患者乳腺癌是否复发，删去含缺失值记录后的数据集中有196条记录没有复发，有81条记录乳腺癌复发。数据集中部分特折与 `variables.xlsx` 中提供的标准值存在差异，为了使结果更加直观，我们更改了记录错误的特征值。

该数据集中的特征多为离散的特征，为了便于后续的分类任务，我们用数字索引替换了每一个特征不同的取值。数字索引与具体特征的对应关系记录在字典 `ind2val` 和 `val2ind` 中。

3.2 数据集划分

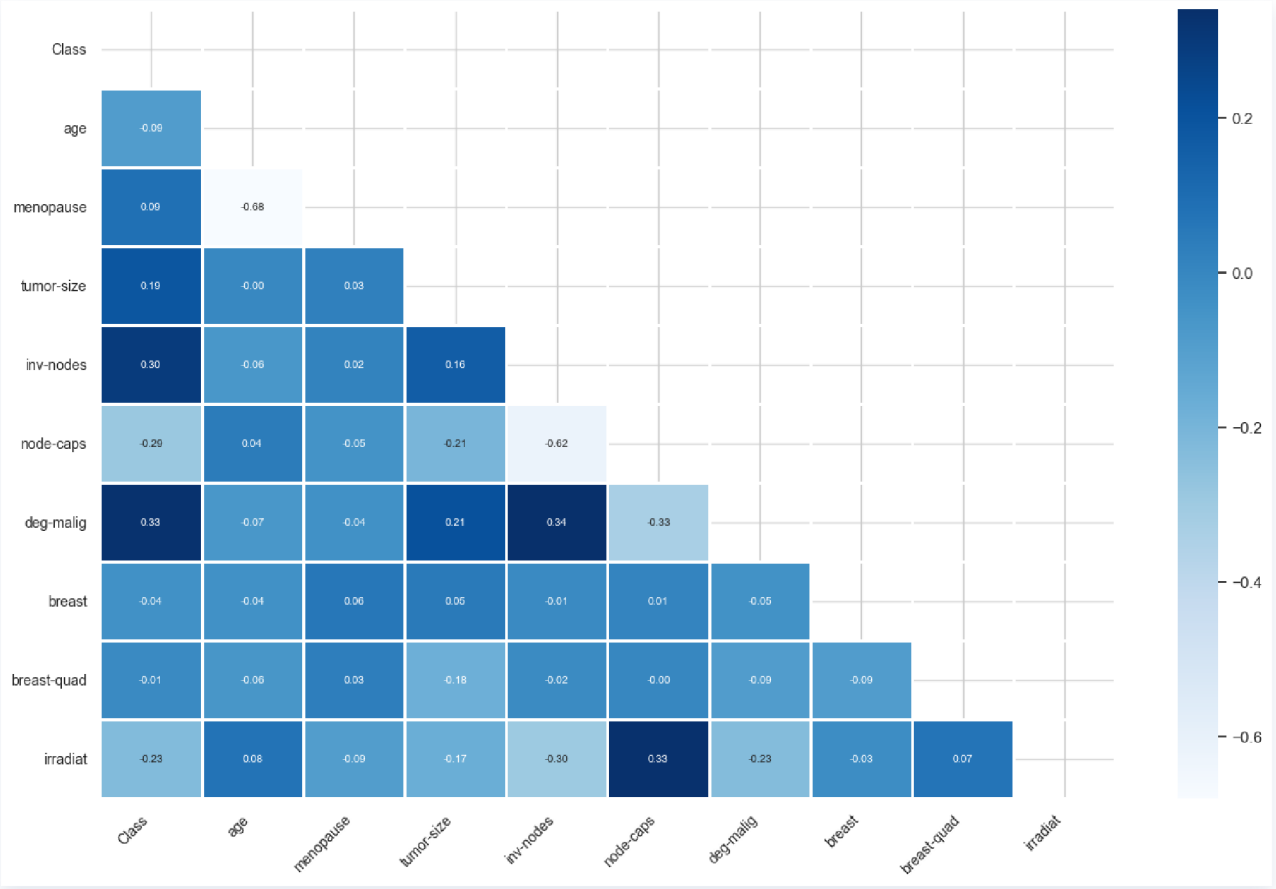
在实验4 Class 的预测任务中，我们仍然遵循前面的范式，按8：2的比例划分训练集和测试集。在训练集中使用 k折交叉验证确定模型较优的超参数。

3.3 分类算法模型

我们采用和实验3预测任务中相同的随机森林分类器模型和支持向量机分类器模型，模型信息在 2.3已有详细的叙述，在此我们不做赘述。

3.4 特征选择与处理

原数据集得到的关联矩阵如下所示：



我们发现该数据集中没有高度关联的特征，因此我们不需要考虑数据去重。为了挑选出较为合适的特征，提高模型的泛化能力，我们采用 `sklearn.feature_selection` 中的 `SelectKBest` 来选择与 `Class` 关联最密切的特征。得到特征的重要性排名为 tumor-size, inv-nodes, deg-malig, age, node-caps, irradiat, menopause, breast, breast_quad。重要性排名和关联矩阵中与 Class 关联值的绝对值大小也有一定的相关性。

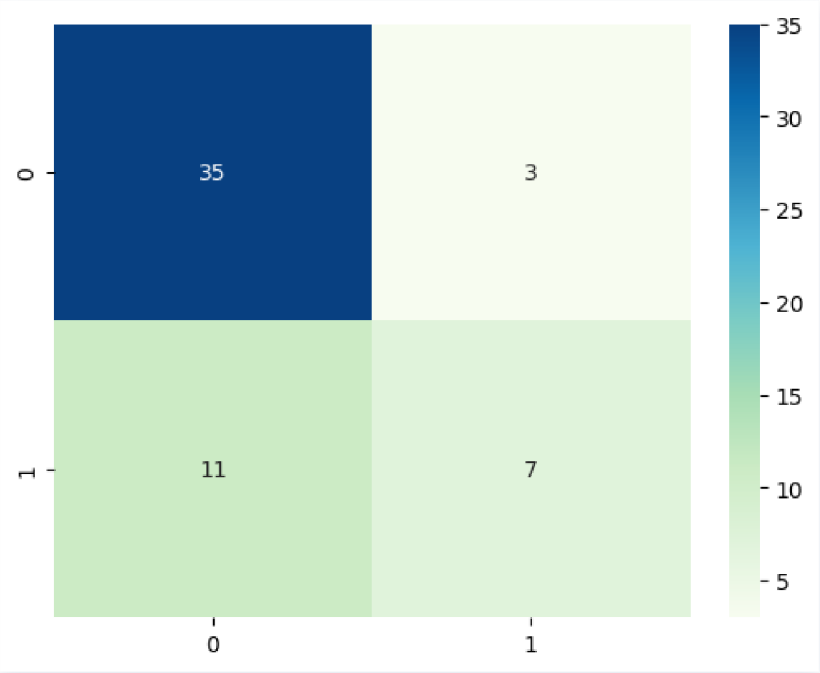
这里我们观察到特征 breast_quad 与 Class 关联不强，因此我们舍去该特征。

3.5 主试验

3.5.1 随机森林

我们使用 `sklearn` 中的 `RandomForestClassifier` 类，设置 `random_state=0` 便于结果复现，其余超参数均保持默认值。我们在训练集和测试集上 Accuracy 分别为 0.964，0.750。其他指标与混淆矩阵如下所示：

指标	Class 0	Class 1
Precision	0.76	0.70
Recall	0.92	0.39
F1-Score	0.83	0.50
Support	38	18

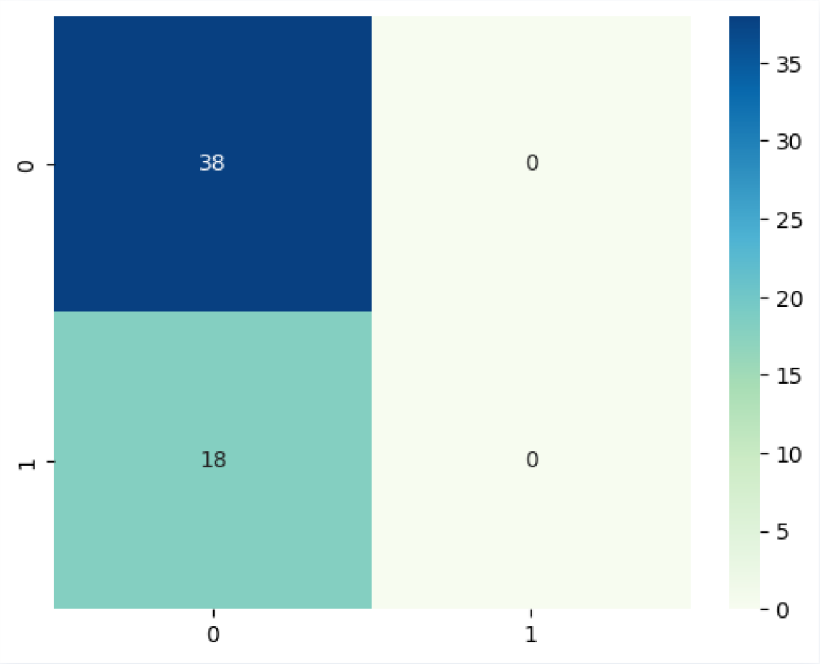


我们的模型在训练集上表现比较优秀，但在测试集上表现较为一般。由上表的评价指标以及混淆矩阵，我们发现我们模型的泛化能力较弱，可能与数据量过小导致训练集数量较小有关。

3.5.2 支持向量机

我们调用 `sklearn` 中的 `SVC` 类来实现分类预测，设置 `random_state=0, probability=True`，其余超参数保持默认。我们在训练集和测试集上 Accuracy 分别为 0.715，0.679。其他指标与混淆矩阵如下所示：

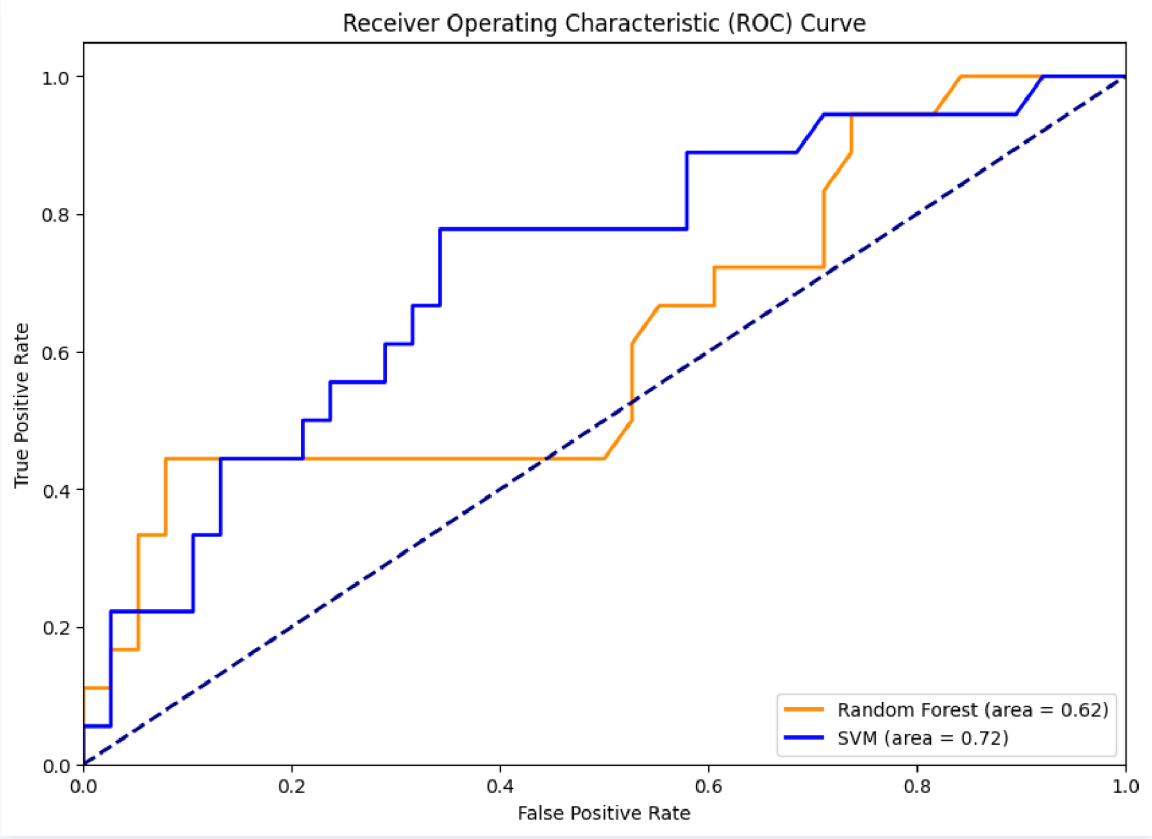
指标	Class 0	Class 1
Precision	0.68	0
Recall	1.00	0
F1-Score	0.81	0
Support	38	18



由 Class 为1的 Precision， Recall以及F1-Score等指标，模型在预测标签为1的样本时表现很差。所有实际标签为1的样本都被错误地预测为0。模型学习效果较差。我们认为在该任务情景下，支持向量机的分类效果并不理想。

3.5.3 模型对比

随机森林和支持向量机在该任务上的 ROC-AUC 图像如下所示：



从图中我们可以得出模型的泛化能力较为糟糕。这种现象可能与以下原因有关：

- **类别不平衡**：可能数据集中标签为1的样本数量很少，导致模型更倾向于预测标签为0。

- **需要更多数据**：如果数据量较小，可能导致模型泛化能力差。

3.6 参数实验

在参数实验中我们选取与2.6中相同的超参数列表来进行选择。

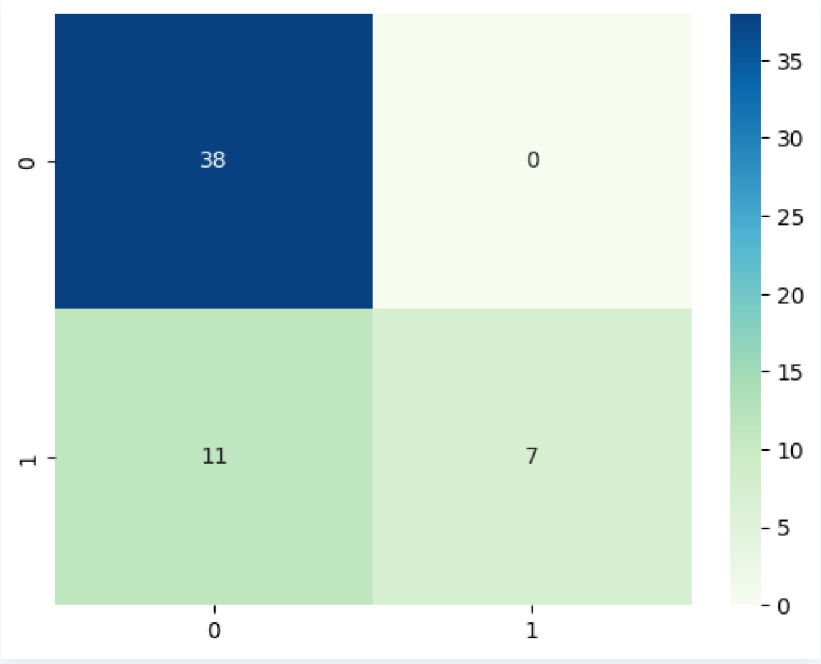
3.6.1 随机森林参数实验

我们获得的一组较优超参数为 'max_depth': None, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 20。使用该超参数，我们在测试集上达到 0.786 的准确率，较默认超参数准确率有小幅提升。

3.6.2 支持向量机参数实验

在支持向量机上我们使用网格搜索，取到的一组较优超参数为 'C': 10, 'gamma': 0.01, 'kernel': 'rbf'。使用该超参数，支持向量机在测试集上达到了 0.804 的准确率，和原始超参数相比，在测试集上的准确率有 18.4% 的提升。此时相关指标与混淆矩阵如下图所示：

指标	Class 0	Class 1
Precision	0.78	1.00
Recall	1.00	0.39
F1-Score	0.87	0.56
Support	38	18



我们发现在新的超参数下 Class 1 的Recall，F1-Score等多项指标均有所改善。

4. 实验总结

本次实验中我们通过随机森林和支持向量机两种分类器模型对数据集进行了分类预测，通过多种指标评判了模型的性能，并通过参数实验调整超参数提高分类预测的综合性能。收获如下：

- 设置训练集，测试集和验证集来对模型进行训练和超参数调整；
- 通过设置随机数种子使得实验可复现；
- 通过Recall，F1-Score和 ROC-AUC 等评判指标加深了对数据本身和模型的理解；
- 理解了随机森林和支持向量机算法。