摘要

本文主要利用 569 个样本的乳腺癌细胞数据，通过探索性数据分析、数据可视化、特征选择、建立模型、模型调参以及模型评估等基本步骤，实现将随机观察到的乳腺癌细胞分类为良性或恶性。以此协助医生提升诊断的效率，减少误诊和漏诊。

数据集展示了良性、恶性的诊断结果与细胞核特征值大小之间的相关性。其中 30 个数值变量代表着每个细胞核的 10 个特征，并计算每个特征的平均值（mean）、标准误差（se）和最大值（worst）。经观察，我们猜测特征之间可能存在严重的多重共线性，并经过数据可视化等方法验证了这一猜想。为减轻多重共线性的对分类结果的影响，我们采用了三种方法进行特征选择，分别为方差膨胀因子检验（VIF）、单变量特征选择以及递归特征消除（RFE），并使用了随机森林和支持向量机进行模型的训练与预测，最后利用 K 折交叉检验进行多角度验证。

综上所述，我们从多角度、多层次分析了乳腺癌疾病诊断的二分类问题。评价模型经过比较全面的性能评价，证明准确率较高。

关键词：特征选择 随机森林 支持向量机

1 模型建立、求解与检验

1.1 问题 1 模型的建立、求解和检验

1.1.1 思路流程

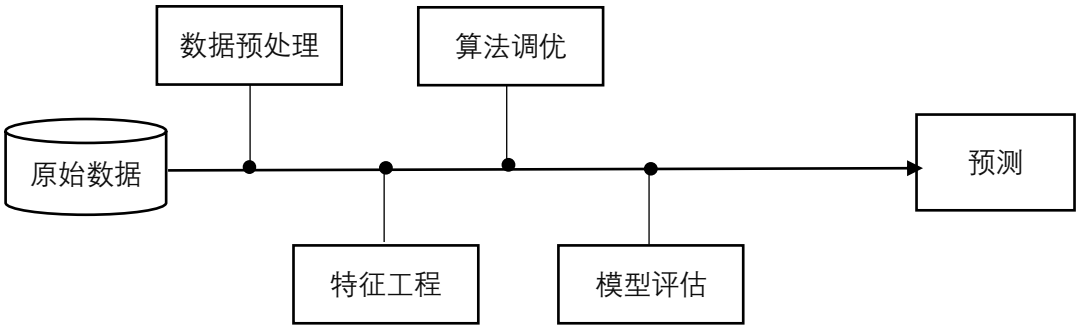

图 1 思路图

1.1.2 探索性数据分析（EDA）

数据集的大小为 569 rows x 32 columns，即一共包含 569 个样本，每个样本包含 32 个属性。其中 357 个（62.7%）被标记为恶性，其余 211 个（37.3%）被标记为良性，结果变量分布图见图 2。

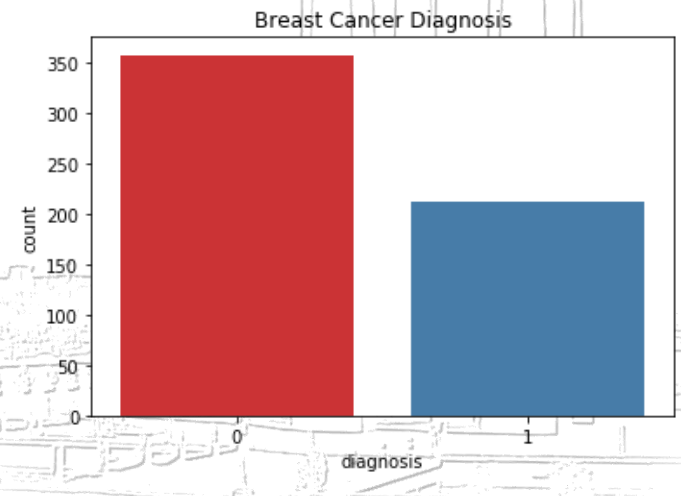

图 2 结果变量分布图

通过数据集的**描述性统计**我们可以发现，除了 ID 号之外，数据集有 30 个数值变量和 1 个分类变量（即目标变量） 其中 30 个数值变量代表着每个细胞核的 10 个特征。此外，还计算了每个特征的**平均值(mean)**、**标准误差(se)**和**最大值(worst)**，数据集中除目标变量外共有 10 x 3 = 30 个特征（列），变量基本信息见表 1，特征表见表 2。

表 1 变量基本信息表

变量名称	变量含义	数据类型
ID number	样本序号	int64
diagnosis	M = malignant 恶性，B = benign 良性	object
radius	半径，细胞核从中心到周边的距离	float64
texture	纹理（灰度值的标准偏差）	float64
perimeter	细胞核周长	float64
area	细胞核面积	float64
smoothness	平滑度（半径长度的局部变化）	float64
compactness	紧凑度（周长^2/面积-1.0）	float64
concavity	凹度（轮廓凹部的严重程度）	float64
concave points	凹点（轮廓凹部的数量）	float64
symmetry	对称性	float64
fractal dimension	分形维数-1	float64

表 2 特征表

特征类别	特征名称	特征含义
radius	radius_mean	半径，细胞核从中心到周边的距离，平均值
	radius_se	半径，细胞核从中心到周边的距离，最大值
	radius_worst	半径，细胞核从中心到周边的距离，标准差
texture	texture_mean	纹理（灰度值的标准偏差），平均值
	texture_se	纹理（灰度值的标准偏差），标准差
	texture_worst	纹理（灰度值的标准偏差），最大值
perimeter	perimeter _mean	细胞核周长，平均值
	perimeter _se	细胞核周长，标准差
	perimeter _worst	细胞核周长，最大值
area	area_mean	细胞核面积，平均值
	area_se	细胞核面积，标准差
	area_worst	细胞核周长，最大值
smoothness	smoothness _mean	平滑度（半径长度的局部变化），平均值
	smoothness _se	平滑度（半径长度的局部变化），标准差
	smoothness _worst	平滑度（半径长度的局部变化），最大值
compactness	compactness_mean	紧凑度（周长^2/面积-1.0），平均值
	compactness_se	紧凑度（周长^2/面积-1.0），标准差
	compactness_worst	紧凑度（周长^2/面积-1.0），最大值
concavity	concavity_mean	凹度（轮廓凹部的严重程度），平均值
	concavity_se	凹度（轮廓凹部的严重程度），标准差
	concavity_worst	凹度（轮廓凹部的严重程度），最大值
concave points	concave points_mean	凹点（轮廓凹部的数量），平均值
	concave points_se	凹点（轮廓凹部的数量），标准差
	concave points_worst	凹点（轮廓凹部的数量），最大值
symmetry	symmetry_mean	对称性，平均值
	symmetry_se	对称性，标准差
	symmetry_worst	对称性，最大值
fractal_dimension	fractal_dimension_mean	分形维数-1，平均值
	fractal_dimension_se	分形维数-1，标准差
	fractal_dimension_worst	分形维数-1，最大值

仅凭特征的名称，我们就可以预见多重共线性的问题，这在后面的部分将被重点讨论。

首先，对数据进行基本的预处理。

（1） 缺失值处理

数据的缺失主要包括记录的缺失和记录中某个字段信息的缺失，两者都会造成分析结果的不准确。对缺失值的处理一般包括删除元组、数据补齐和不处理。编写 missing_value_table 函数检查数据集是否存在缺失值。

```
# function to calculate missing values by column
def missing_values_table(data):
    # total miss values
    mis_val = data.isnull().sum()

    # percentage of missing values for each columns
    mis_val_percent = 100*data.isnull().sum()/len(data)

    # make a table with the results
    mis_val_table = pd.concat([mis_val, mis_val_percent], axis=1)

    # rename the columns
    mis_val_table_ren_columns = mis_val_table.rename(
        columns = {0: 'Missing Values', 1: '% of Total Values'})

    # sort the table by percentage of missing descending
    mis_val_table_ren_columns = mis_val_table_ren_columns[
        mis_val_table_ren_columns.iloc[:, 1]!=0].sort_values(
        '% of Total Values', ascending=False).round(1)


    # print some summary infomation
    print("You selected dataframe has "+str(data.shape[1]) +
          " columns.\nThere are " + str(mis_val_table_ren_columns.shape[0])+
          " columns that have missing values.")

    # return the dataframe with missing infomation
    return mis_val_table_ren_columns
```

```
You selected dataframe has 32 columns.
There are 0 columns that have missing values.
```

Out[6]:

Missing Values	% of Total Values

输出结果显示，数据集完整，**不存在缺失值**，所以无需进行下一步处理。

（2） 删除无关变量

由于 ID number 列并未向我们提供任何有关癌细胞分类的信息，应当将其删除。

```
data.drop(['ID number'], axis=1, inplace=True)
```

（3） 处理分类变量

根据分类标签值 M 和 B 可以知道，（M）意味着诊断结果为恶性，（B）意味着诊断结果为良性。为方便后续模型的训练与评估，我们分别将这些值映射到 1 和 0 。

```
data['diagnosis'] = data['diagnosis'].map({'M':1, 'B':0})
```

（4） 标准化数据

通过观察数据的数量级可以发现数据之间存在**量纲的差距**，例如，area_mean 的最大值为 2501，而 smoothness_mean 的最大值只有 0.163400。因此，在可视化、特征选择之前，我们需要将数据进行标准化，使其遵循标准高斯分布。

```
1. original_features = data.drop('diagnosis', axis = 1)
2. standard_features = (original_features - original_features.mean()) / origina
   l_features.std()
3. standard_data = pd.concat([data['diagnosis'], standard_features], axis = 1)
```

1.1.3 数据可视化（Visualization）

进一步观察数据的基本信息之后可发现，虽然数据集提供了 30 个特征，但仅凭特征本身的名称，我们就可以预见特征之间可能存在多重共线性。例如 Radius 半径，Perimeter 周长 与 Area 面积等，可以类比基本图形圆，见表 3。

表 3 圆

	计算公式
半径	R
周长	$2\pi R$
面积	πR^2

通过计算 radius_mean 与 perimeter_mean 这个两个变量之间的皮尔逊（Pearson）

相关系数来初步验证之前的猜想。

```
from scipy.stats import pearsonr
r, p = pearsonr(data["radius_mean"],data["perimeter_mean"] )
```

最终得到 r 为 0.9978552814938109，p 为 0.0。r 趋近 1，说明两个变量呈现很强的正相关性，猜测存在合理性。

由于数据集的 30 列都包含关于相同 10 个关键属性的信息，但它们的视角不同（即 mean、se 和 worst）。为更好地观察，可以尝试仅通过其中一个角度来分析数据从而挖掘出一些有用的信息。这里将特征分为 3 组，feature_mean、feature_se、feature_worst，选取 feature_mean 进行可视化以进一步探索特征之间的关系。feature_mean 的前 5 rows x 5 columns 数据见表 4。

```
feature_mean = data.iloc[:, 1:11]
feature_se = data.iloc[:, 11: 21]
feature_worst = data.iloc[:, 21:31]
```

表 4 feature_mean 的部分数据信息表

radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
17.99	10.38	122.80	1001.0	0.11840
20.57	17.77	132.90	1326.0	0.08474
19.69	21.25	130.00	1203.0	0.10960
11.42	20.38	77.58	386.1	0.14250
20.29	14.34	135.10	1297.0	0.10030

（1） 小提琴图（Violin Plot）和箱型图（Box-plot）

小提琴图（Violin Plot）是用来展示多组数据的分布状态以及概率密度。这种图表结合了箱形图和密度图的特征，主要用来显示数据的分布形状。跟箱形图类似，但是在密度层面展示更好。在数据量非常大不方便一个一个展示的时候小提琴图特别适用。小提琴的概念图见图 3。

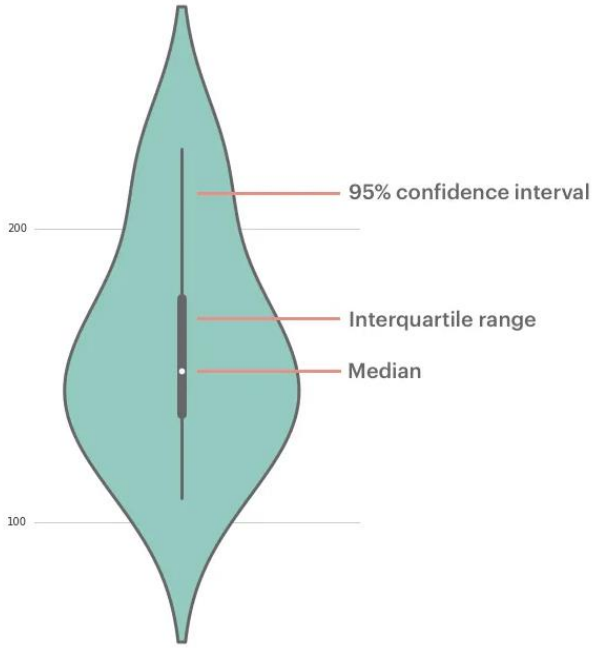

图 3 小提琴图概念图

```
data = pd.concat([data["diagnosis"],feature_mean],axis=1)
data = pd.melt(data,id_vars="diagnosis",
                    var_name="features_mean",
                    value_name='value')
plt.figure(figsize=(6,6))
sns.violinplot(x="features_mean", y="value", hue="diagnosis", data=data,split=True, inner="quart",palette="Set1")
plt.xticks(rotation=45)
```

箱型图（Box-plot）又称为盒须图、盒式图或箱线图，是一种用于展示一组数据分散情况的统计图，因形状如箱子而得名，它能显示出一组数据的最大值，最小值、中位数及上下四分位数。

```
plt.figure(figsize=(6,6))
sns.boxplot(x="features_mean", y="value", hue="diagnosis", data=data,palette="Set1")
plt.xticks(rotation=45)
```

feature_mean 的小提琴图和箱型图分别见图 4，图 5。

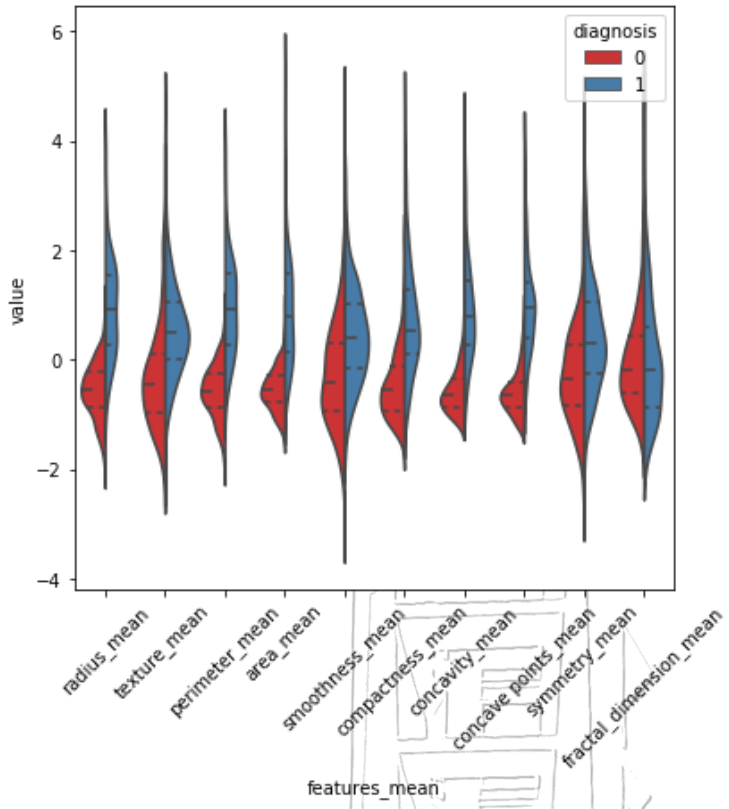

图 4 **feature_mean** 的小提琴图

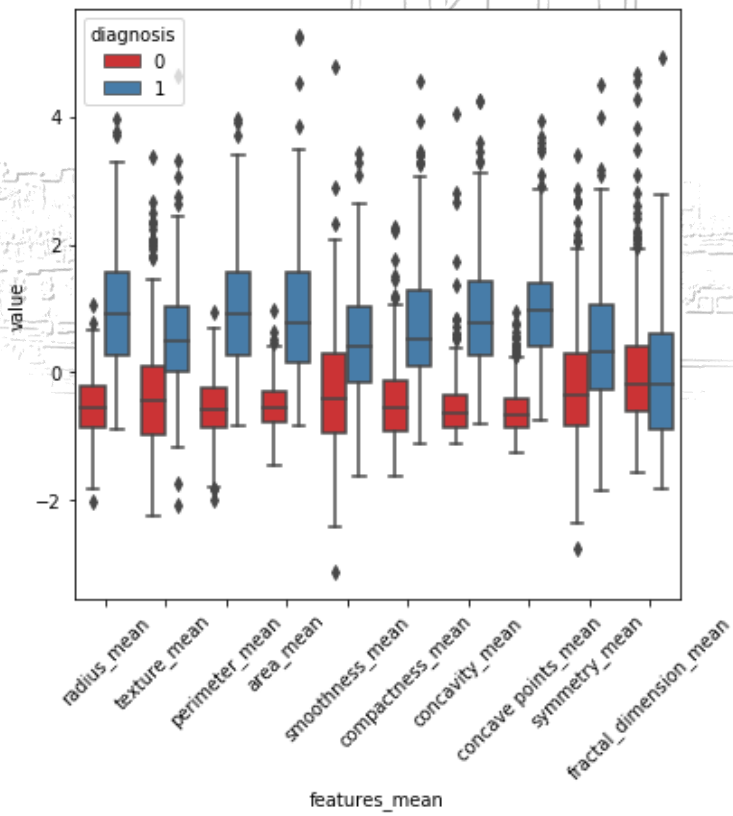

图 5 **feature_mean** 的箱型图

（2） 分簇散点图（Swarm Plot）

分簇散点图（Swarm Plot）可以自己实现对数据分类的展示，也可以作为箱型图或小提琴图的一种补充，用来显示所有结果以及基本分布情况。可以更加直观地看出特征对分类结果的影响程度。feature_mean 的分簇散点图见图 6。

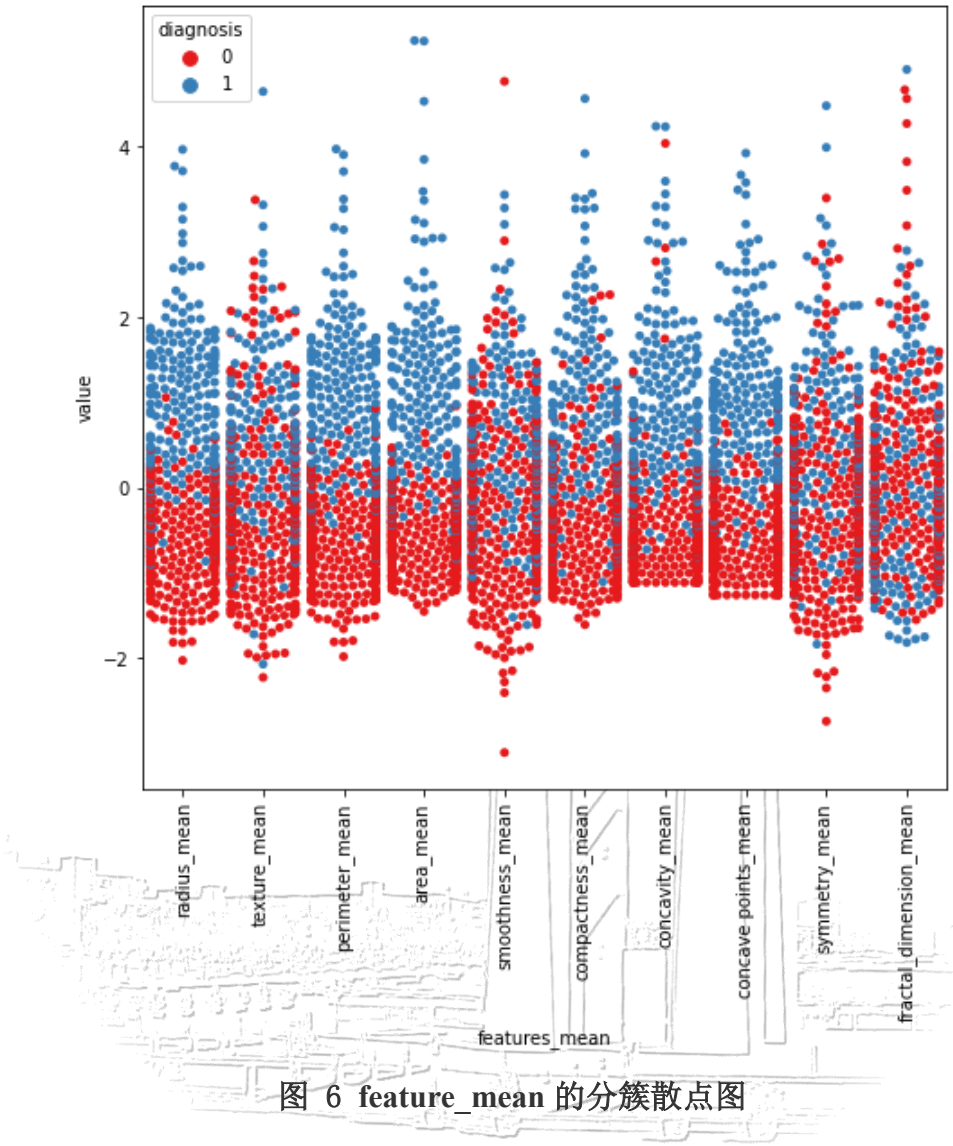

图 6 feature_mean 的分簇散点图

从这些图中都可以清楚地看出，有部分特征对结果的分类起到了很大作用，而有部分特征并不能提供很好的分类信息。例如， texture_mean 这个特征中，恶性和良性的中位数看起来是分开的，所以可以很好地进行分类。然而，在 fractal_dimension_mean 这个特征中，恶性和良性的中位数看起来并没有分离，所以不能提供很好的分类信息。这意味着，合理的特征选择对尤为重要。

（3） 散点图矩阵（Pairs Plot）

散点图矩阵允许同时看到多个单独变量的分布和它们两两之间的关系。

feature_mean 的散点图矩阵如图 7。

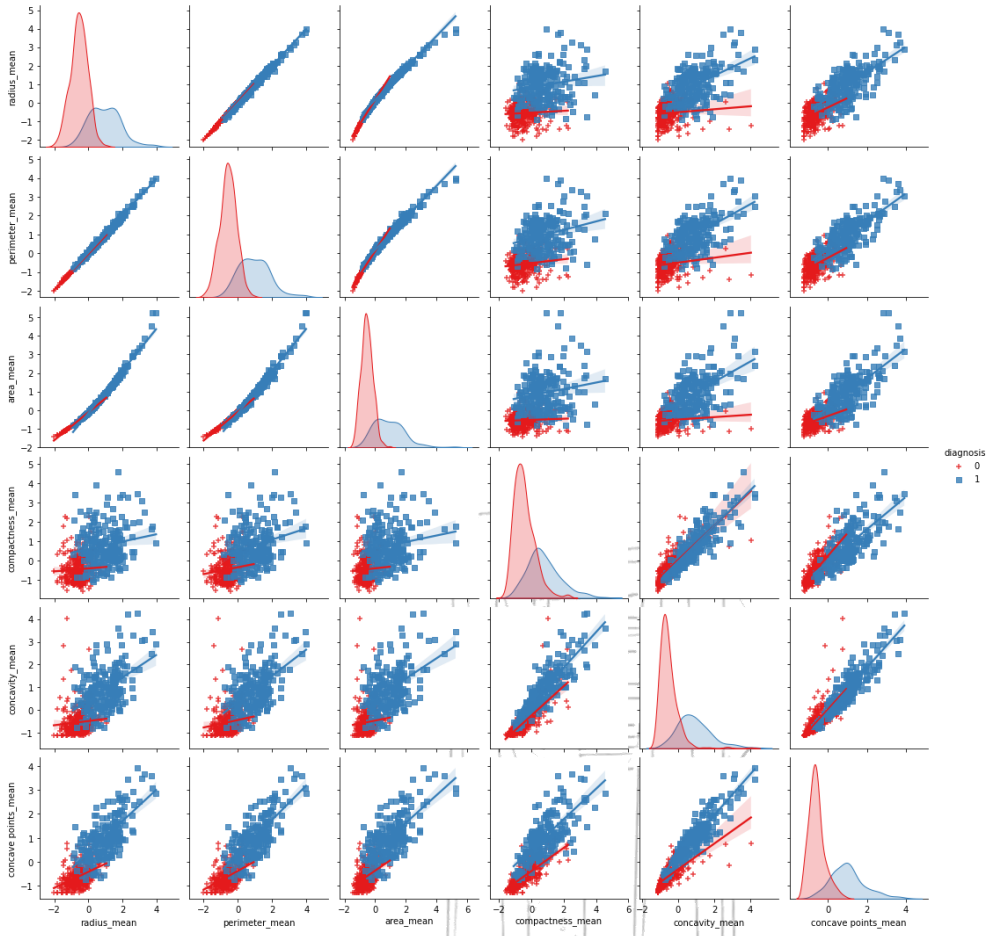

图 7 feature_mean 的散点图矩阵

在散点图矩阵中有一些有趣的图案可见。

例如，radius、perimeter 和 area 属性之间几乎完美的线性关系暗示着这些变量之间存在多重共线性。concavity、concave_points 和 compactness 之间也可能存在多重共线性。

（4） 绘制热力图（HeatMap）

热力图通过皮尔逊相关系数来查看变量之间的关联性，见图 8，此图验证了我们之前的猜测，我们的数据存在严重的多重共线性（高度相关）问题。我们可以观察到以下特征相互呈正相关：

radius、perimeter、area、compactness、concavity、concave points

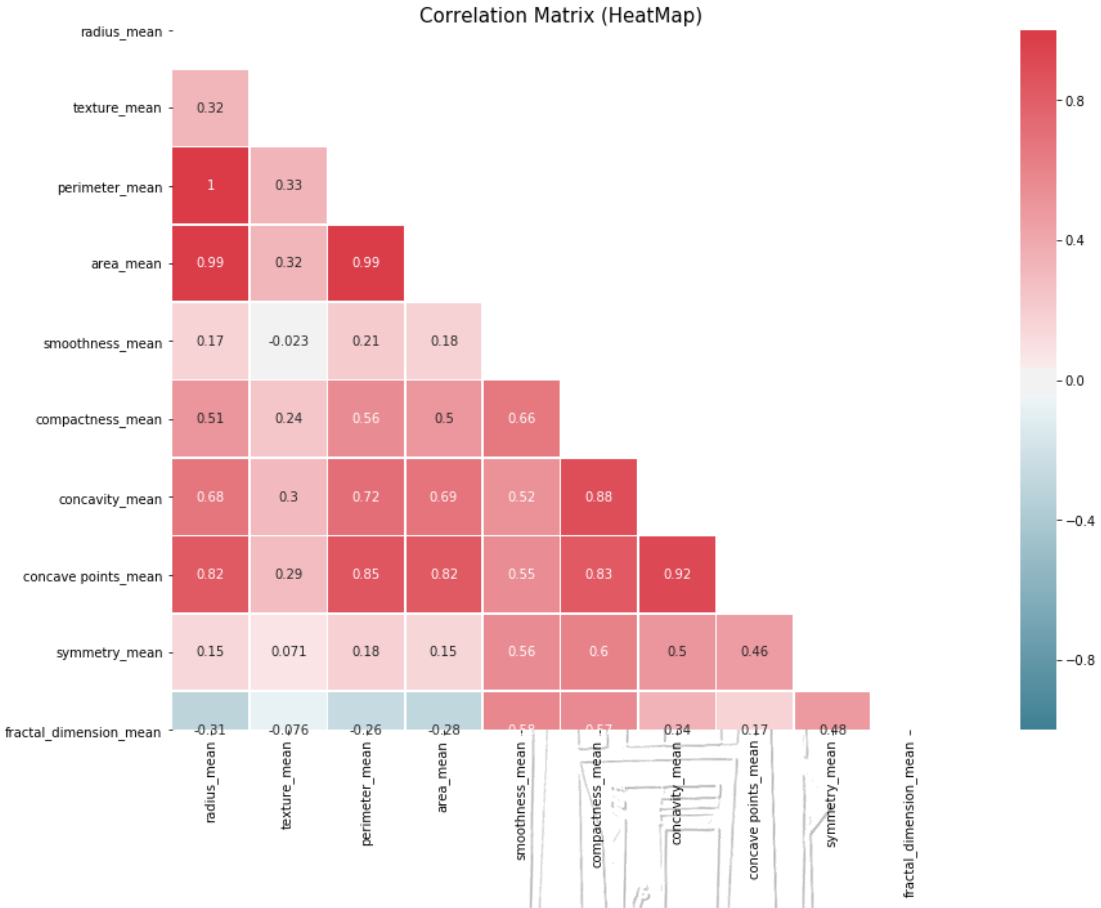

图 8 feature_mean 的热力图

1.1.4 特征选择（Feature Selection）

特征选择是指将高维空间的样本通过映射或者是变换的方式转换到低维空间，达到降维的目的，然后通过特征选取剔除冗余和不相关的特征来进一步降维。进行特征选择的原因主要是在有限的样本数目下，用大量的特征来设计分类器计算开销太大而且分类性能差。

特征选择的原则是获取尽可能小的特征子集，不显著降低分类精度，不影响类分布以及特征子集应具有稳定适应性强等特点。这里采用三种不同的方式进行特征选择。

（1）方差膨胀因子（VIF）检验

VIF 是衡量模型中解释变量多重共线性严重程度的一种度量。变量多重共线性越严重，VIF 越大。

$$VIF = \frac{1}{1 - R^2}$$

式中，R^2 是决定系数。

```
1.  # 定义一个计算 VIF 的函数
```

```
2.
3.  def calculate_vif(df):
4.      vif = pd.DataFrame()
5.      vif['Feature'] = df.columns
6.      vif['VIF'] = [variance_inflation_factor(df.values, i) for i in range(df.
    shape[1])]
7.      return (vif)
8.
9.  # 构造 VIF 矩阵
10. from statsmodels.stats.outliers_influence import variance_inflation_factor
11. from sklearn.feature_selection import VarianceThreshold, SelectKBest, chi2,
    RFE, RFECV
12. vif_table = calculate_vif(original_features)
13. vif_table = vif_table.sort_values(by = 'VIF', ascending = False)
14. vif_table
15.
16. # 选出 VIF 最高的 5 个变量并且进行剔除
17.
18. features_to_drop = list(vif_table['Feature'])[:5]
19. new_features = data.drop(features_to_drop, axis = 1)
```

计算出各变量 VIF 的值，并将 Top5 进行剔除，用剩下的特征作为最终的特征，见表 4。

表 2 VIF TOP5

Rank	Feature	VIF
1	radius_mean	63306.172036
2	perimeter_mean	58123.586079
3	radius_worst	9674.742602
4	perimeter_worst	4487.781270
5	area_mean	1287.262339

（2）单变量特征选择

单变量特征选择的原理是分别单独计算每个变量的某个统计指标，根据该指标来判断哪些指标重要，剔除那些不重要的指标。

使用 SelectKBest 中的卡方（chi2）检验方法选出前 15 个与标签最相关的特征，以这 15 个特征作为最终选择的特征，见表 5。

```
1.  # 实例化
2.  select_features = SelectKBest(chi2, k = 15).fit(X_train, Y_train)
3.
```

```
# Top 15
selected_features = select_features.get_support()
print("Top 15 features: ", list(X_train.columns[selected_features]))
```

表 5 单变量特征选择 TOP5

Rank	Feature
1	radius_mean
2	texture_mean
3	perimeter_mean
4	area_mean
5	concavity_mean
6	radius_se
7	perimeter_se
8	area_se
9	radius_worst
10	texture_worst
11	perimeter_worst
12	area_worst
13	compactness_worst
14	concavity_worst
15	concave points_worst

（3）递归特征消除

递归特征消除法是通过递归减少考察的特征集规模来选择特征。首先，预测模型在原始特征上训练，每个特征指定一个权重，之后，那些拥有最小绝对值权重的特征被剔除。如此往复递归，直至剩余的特征数量达到所需要的特征数量。

采用集成算法中的随机森林方法作为底层模型，选出前 15 个特征，见表 6。

```
# 实例化以选择前 15 个特征
rf = RandomForestClassifier(random_state = 42)
rfe = RFE(estimator = rf, n_features_to_select = 15).fit(X_train, Y_train)

# Top 15 features
print("Top 15 features: ", list(X_train.columns[rfe.support_]))
```

表 6 RFE TOP5

Rank	Feature
1	radius_mean
2	texture_mean
3	perimeter_mean
4	area_mean
5	concavity_mean
6	radius_se
7	perimeter_se
8	area_se
9	radius_worst
10	texture_worst
11	perimeter_worst
12	area_worst
13	compactness_worst
14	concavity_worst
15	concave points_worst

1.1.5 模型选择（Model Selection）

通过观察数据可知，附件 1 中，62.7% 被标记为恶性，其余 37.3% 被标记为良性 ，两者比例约为 2：1，数据存在一定的不平衡性，见 Error! Reference source not found.。使用逻辑回归、决策树或神经网络模型，容易产生过拟合现象；而使用过采样、欠采样或综合采样方法消除数据的不平衡性，又容易造成数据的失真，不易对每家企业做出较为准确的信贷风险评估。

基于此，本文采用支持向量机模型和随机森林模型。

其中，支持向量机模型其基本思想是：通过寻求结构化风险最小来提高学习机泛化能力，支持向量机的学习策略便是间隔最大化，即求解能够正确划分训练数据集并且几何间隔最大的分离超平面，见图 9。从而达到在统计样本量较少的情况下，亦能获得良好统计规律的目的。

随机森林模型其基本思想是：构建多个独立的决策树模型，然后在样本上训练树模型，训练结束后，通过对所有树模型取加权平均确定最终分类结果。随机森林模型的优点在于能够在较高的正确率下，很好地避免过拟合性，比较适合处理不平衡的数据。将“是否为良性”作为因变量，上述分类特征作为自变量，进行模型的训练与拟合。

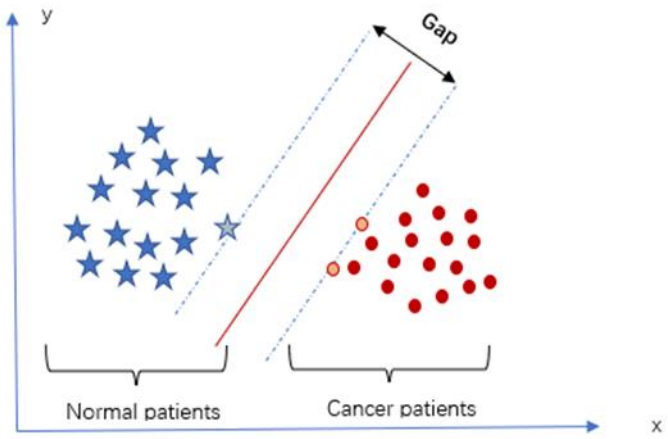

图 9 间隔最大化

1.1.6 模型调参（Tuning hyperparameters）

将数据拆分为训练数据集和验证数据集，使用训练数据集进行模型调参和预测，使用验证数据集进行模型的验证。

```
X_train , X_test , y_train , y_test = train_test_split(
    X_data,
    y_data,
    train_size = 0.75,
    random_state = 2607
    )
```

A. 支持向量机模型

运用网格搜索方法，对随机森林的 C， gamma， kernel 参数进行调整，以 AUC（Area Under Curve）作为模型性能的指标，选择出能使模型表现最好的参数，见表 。

```
from sklearn.model_selection import GridSearchCV

empty = SVC()
params = {"C":[0.001,0.01,0.1,1,10,100],
              "kernel":["rbf","linear"],
              "gamma":[0.001,0.01,0.1,1,10,100]
          }
Grid = GridSearchCV(empty,params,refit=True,scoring='accuracy').fit(X_train,
y_train)
Grid.best_params_
```

表 7 模型最优参数表

参数名称	最优参数值	参数含义
C	100	控制损失函数的惩罚系数
gamma	0.001	核函数系数
kernel	linear	算法中采用的核函数类型

B. 随机森林模型

运用网格搜索方法，对随机森林的 max_depth, n_estimators, max_features, min_sample_split 参数进行调整，以 AUC(Area Under Curve)作为模型性能的指标，选择出能使模型表现最好的参数，见表 8。

```
from sklearn.model_selection import GridSearchCV

rfc = RandomForestClassifier()
skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=17)
rfc_params = {'max_features': range(1,11),
            'min_samples_leaf': range(1,3),
            'max_depth': range(3,13),
            'criterion':['gini','entropy']}
gcv = GridSearchCV(rfc, rfc_params, n_jobs=-1, cv=skf, scoring='recall')
gcv.fit(X_train, y_train)
```

表 8 模型最优参数表

参数名称	最优参数值	参数含义
max_depth	8	森林中每棵决策树的深度
n_estimators	100	决策树个数
max_features	8	每棵决策树使用的变量占比
min_sample_split	1	叶子的最小拆分样本量

1.1.7 模型评价与预测（Model evaluation and prediction）

将训练数据集代入模型，得到预测值，并与其对应的实际值进行比较。

首先，绘制两种模型的 ROC 曲线(Receiver Operating Characteristic Curve)，并计算 AUC 面积见图 、图 11。一般在 AUC 面积在 0.8 以上的模型性能较好，因此选用的两种模型的 AUC 面积指标较好。

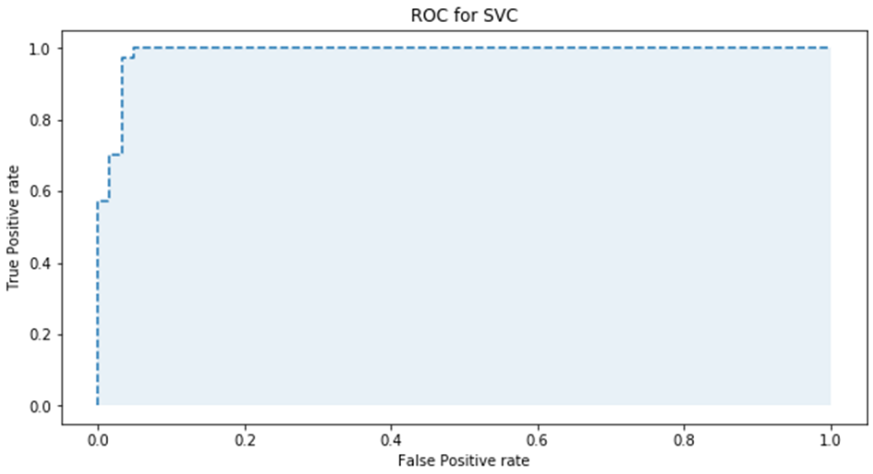

图 10 支持向量机模型 ROC 曲线

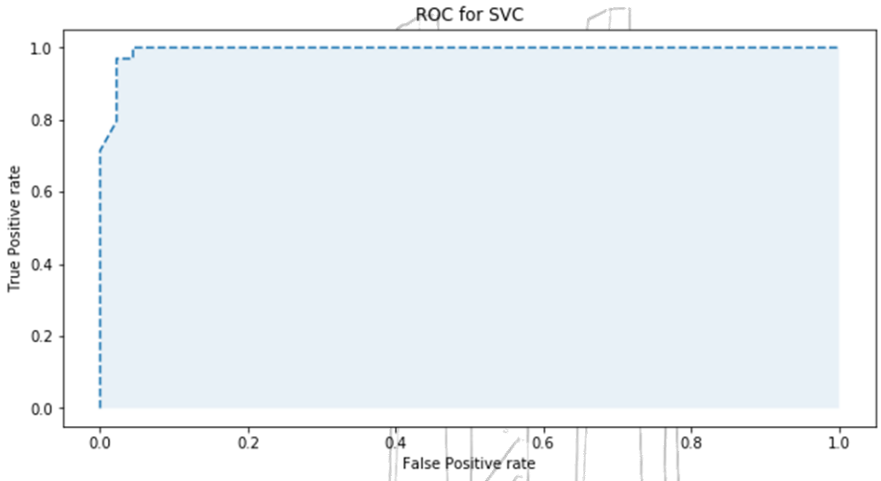

图 11 随机森林模型 ROC 曲线

其次，对两个模型的准确率进行评价。

A. 支持向量机模型

其中，支持向量机模型计算得预测细胞为良性的准确率达到 100%，预测细胞为恶性的准确率达到 97%，整体准确率达到 98%，模型预测结果相对较好。

表 9 支持向量机模型准确率

	准确率/%	计算方法
预测良性	1.0	预测良性正确的数量除以良性样本总数
预测恶性	0.97	预测恶性正确的数量除以恶性样本总数
整体准确率	0.98	正确的数量除以样本总数

B. 随机森林模型

随机森林模型计算得预测细胞为良性的准确率达到 100%，预测细胞为恶性的准确率达到 98%，整体准确率达到 99%，模型预测结果相对较好。

表 10 随机森林模型准确率

	准确率/%	计算方法
预测良性	1.0	预测良性正确的数量除以良性样本总数
预测恶性	0.98	预测恶性正确的数量除以恶性样本总数
整体准确率	0.99	正确的数量除以样本总数

2 模型评价

2.1 优点

（1）通过多种数据挖掘方法，选择能准确反映样本的指标体系，有利于模型准确率的提高；

（2）采用了多种方法进行特征选择，考虑较为全面。

（3）采用了合适的方法进行细胞核为良性或恶性的预测，如支持向量机、随机森林、网格搜索等，使得模型准确率提高较大；

2.2 缺点

（1）模型存在一定的过拟合。