


中国研究生创新实践系列大赛

“华为杯”第十八届中国研究生

数学建模竞赛

学　校		上海理工大学
参赛队号		**21102520357**
队员姓名	**1.**	应旭臣
	2.	严裕
	3.	李淳风

中国研究生创新实践系列大赛

“华为杯”第十八届中国研究生

数学建模竞赛

题 目　　抗乳腺癌候选药物的优化建模

摘　　要：

乳腺癌是目前世界上最常见，致死率较高的癌症之一。有研究发现，ERα 基因在乳腺发育过程中扮演了十分重要的角色。目前，抗激素治疗常用于 ERα 表达的乳腺癌患者，其通过调节雌激素受体活性来控制体内雌激素水平。因此，ERα 被认为是治疗乳腺癌的重要靶标，能够拮抗 ERα 活性的化合物可能是治疗乳腺癌的候选药物。在药物研发中，为了节约时间和成本，通常采用建立化合物活性预测模型（Quantitative Structure-Activity Relationship, QSAR）的方法来筛选潜在活性化合物。本文针对乳腺癌治疗靶标 ERα，以题目给定的 ERα 拮抗剂信息，通过数据挖掘技术来构建化合物生物活性的定量预测模型和 ADMET 性质的分类预测模型。从而为同时优化 ERα 拮抗剂的生物活性和 ADMET 性质提供预测服务。

针对问题一，依据分子描述符与化合物生物活性之间的相关性，对影响分子描述符的相关变量进行重要性排序，筛选出前 20 个具有显著影响的分子描述符。首先对数据进行了**预处理**，减少了 225 个变量数据和 9 组样本数据，其次建立综合筛选模型对预处理后的数据使用**随机森林**的方法对变量的重要性进行排序，再对排名靠前的变量进行**高相关性滤波**处理，剔除几个相关性强的变量，得到重要性排名前 20 个主要变量。经过合理性评价，证明我们挑选出的主要变量具有**代表性和独立性**。

针对问题二，利用问题一所筛选出的 20 个主要变量作为输入，生物活性值 pIC_{50} 作为输出，利用分别建立了基于**多元线性回归**，**支持向量机**，**高斯过程回归**和 BP 神经网络等多种**预测模型**，并对预测结果对比，最终得出基于**遗传算法优化**的 BP 神经网络预测效果最好。并利用训练得到的神经网络完成了对 test 集生物活性的预测。

针对问题三，对数据集中的 729 个分子描述符，和 1974 个化合物的 ADMET 数据分别构建分类预测模型，即**构建五个二分类预测模型**。我们首先对原始数据打乱分割，选择该数据集中的 80%数据作为训练集，20%数据作为测试集。在本题中，我们基于 **XGBOOST 框架**构建 **GBDT 算法**，使用 python 搭建普适性的**二元-逻辑斯蒂克（Binary-Logistic）分类器**脚本。在决策树的分裂查找算法上，采用**精确贪心算法**枚举，计算每棵树的**信息增益**作为分裂准

则。此外，根据**交叉验证**结果对其进行调参优化，验证找出分类器模型的最优参数。保存已经构建好的分类器模型后，对不同模型的预测指标其进行可视化比较。根据决策树在计算过程对每个特征变量分裂的次数，评价增益和平均覆盖率，作为特征变量影响目标指标的决策依据，得到不同特征变量影响因素的可视化结果。最终返回值为概率分布结果，根据召回率和特异性的几何平均数（G-Mean）其进行最优阈值选取。最后，针对本题要求，预测结果**按照生化医学检测标准进行可视化输出**，在图表中突出展示检测结果为阳性的样本。

针对问题四，从分子描述符中寻找到合适的区间范围，要求该**区间范围内分子描述符所影响的样本结果同时满足生物活性的要求和 ADMET 性质要求。**我们首先以化合物样本作为研究对象，对其 729 个分子描述变量进行数据降维。数据降维过程分为两步，第一次筛选的降维依据是前两问得到 40 组代表性特征变量。随后对于题目所提出的两点约束要求，本文考虑建立一个合理化的特征指标来对化合物能否满足以上两点进行评估。获得药物性分类指标后，对总体样本进行分类操作，获得多分类问题的目标数据集。构建基于 **XGBOOST** 框架下的**多分类（Multi-Classification）**分类器脚本，测试模型性能指标，计算特征变量之间的 **F-1 score**，得到对模型性能影响较大的特征变量。以多分类预测模型结果，找到同时满足约束的特征变量进行二次筛选，得到目标特征变量。最后利用聚类方法，对目标特征变量进行可视化聚类分析，完成特征变量范围的选取。

关键词：Erα 基因，QSAR 模型，相关性分析，随机森林，遗传算法优化，BP 神经网络，XGBoost 框架

目录

一、问题重述

1.1 问题背景

乳腺癌是目前世界上最常见，致死率较高的癌症之一。据 2018 年国际癌症研究机构（IARC）调查的最新数据显示，乳腺癌在全球女性癌症中的发病率为 24.2%，位居女性癌症的首位，其中 52.9%发生在发展中国家[1]。乳腺癌的发展与雌激素受体密切相关，有研究发现，ERα 基因在乳腺发育过程中扮演了十分重要的角色。目前，抗激素治疗常用于 ERα 表达的乳腺癌患者，其通过调节雌激素受体活性来控制体内雌激素水平。因此，ERα 被认为是治疗乳腺癌的重要靶标，能够拮抗 ERα 活性的化合物可能是治疗乳腺癌的候选药物。比如，临床治疗乳腺癌的经典药物他莫昔芬和雷诺昔芬就是 ERα 拮抗剂。通过去除或阻断激素的作用，以阻止癌细胞生长。与化疗相比，抗激素治疗具有疗效确切、毒性小、使用方便、无须住院、患者易于接受等优点，虽起效慢，但缓解期长，特别适合于激素受体（ER/PR）阳性的各期乳腺癌患者[5]。因此，抗激素治疗药物的研发将给癌症患者带来福音。

在药物研发中，为了节约时间和成本，通常采用建立化合物活性预测模型的方法来筛选潜在活性化合物。具体做法是：针对与疾病相关的某个靶标，收集一系列作用于该靶标的化合物及其生物活性数据，然后以一系列分子结构描述符作为自变量，化合物的生物活性值作为因变量，构建化合物的定量结构-活性关系（QSAR）模型，然后使用该模型预测具有更好生物活性的新化合物分子，或者指导已有活性化合物的结构优化。完善的 QSAR 模型能够确定关键的分子结构符并预测化合物的性质，对已知活性化合物研究提供参考，对结构相似的化合物功能做出合理的预测[2]。

此外，一个化合物想要成为候选药物，除了需要具备良好的生物活性外，还需要在人体内具备良好的药代动力学性质和安全性，合称为 ADMET（Absorption 吸收、Distribution 分布、Metabolism 代谢、Excretion 排泄、Toxicity 毒性）性质。其中，ADME 主要指化合物的药代动力学性质，描述了化合物在生物体内的浓度随时间变化的规律，T 主要指化合物可能在人体内产生的毒副作用。一个化合物的活性再好，如果其 ADMET 性质不佳，比如很难被人体吸收，或者体内代谢速度太快，或者具有某种毒性，那么其仍然难以成为药物，因而还需要进行 ADMET 性质优化。

因此，在筛选一个候选药物时，应该设法充分利用有价值的信息（如，分子描述符），综合考虑其生物活性、药代动力学性质和安全性，建立完善的 QSAR 模型，从而对化合物性质进行合理预测。 虚拟筛选潜在活性化合物有助于生物医学研究人员设计实验方向，对于药物研发有着重要意义[3]。

1.2 问题提出

本文基于化合物的定量结构-活性关系（Quantitative Structure-Activity Relationship, QSAR）模型，针对乳腺癌治疗靶标 ERα，以题目给定的 ERα 拮抗剂信息（1974 个化合物样本，每个样本都有 729 个分子描述符变量，1 个生物活性数据，5 个 ADMET 性质数据），通过数据挖掘技术来建立构建化合物生物活性的定量预测模型和 ADMET 性质的分类预测模型。从而为同时优化 ERα 拮抗剂的生物活性和 ADMET 性质提供预测服务。下面将对该数学问题进行提出和描述。

整个数学问题是递进描述的。参考给出的 ERα 拮抗剂信息，首先需要理解化合物的结构式(SMILES)、生物活性值(pIC_{50})、分子描述符信息以及药代动力学性质和安全性(ADMET)等变量的生物化学意义，并对 1974 个样本数据进行预处理，以降低化合物生物活性过低对整体数据集的影响，提高数据的可用性。在预处理的基础上需要建立降低生物活性定理预测模型，面对几百个化学分子描述符信息符变量的情况下，需要通过找特征值或者降维的方法处理，筛选出对生物活性影响的主要影响因素。之后针对多变量，非线性的复杂问题，利用数据挖掘和机器学习技术建立化合物生物活性的定量预测模型和 ADMET 性质提供预测服务，并验证模型的准确性，优化得到一般性的数学模型，从而对测试集的化合物活性和 ADMET 数据进行定理预测。进一步的，需要综合考虑化合物生物活性、药代动力学性质和安全性，寻找并给出合理的分子描述符，并确定他们的取值范围。最终，将优化方案可视化展示，为药物研发企业提供可靠有效的乳腺癌治疗靶标 Erα 抑制化合物的预测模型和优化方法，给癌症患者带来福音。

本文需要完成以下几个问题：

问题一：不同分子描述符对化合物生物活性影响的重要性影响不同。合理的分子描述符变量选择，对建立生物活性的定量预测模型有着重要影响。根据分子描述符与化合物生物活性之间的相关性，进行重要性进行排序，筛选出前 20 个最具有显著影响的分子描述符（即变量），并详细说明分子描述符筛选过程及其合理性。问题一主要是针对附件 1 和附件 2 中的 1974 个化合物的 729 个分子描述符进行主要变量筛选。

问题二：结合问题 1 选择的重要描述符变量，建化合物对 Erα 生物活性的定量预测模型。然后使用构建的预测模型，对附件 1 的测试集化合物活性进行预测，并填入表格的 IC50_nM 列及对应的 pIC50 列中。

问题三：要求针对附件 2 和附件 4 中的 1974 个化合物的 ADMET 数据，分别构建化合物的 Caco-2、CYP3A4、hERG、HOB、MN 的分类预测模型。之后使用所构建的 5 个分类预测模型，对文件“ADMET.xlsx”的 test 表中的 50 个化合物进行相应的预测，并将结果填入“ADMET.xlsx”的 test 表中对应的 Caco-2、

CYP3A4、hERG、HOB、MN 列。

问题四：寻找并阐述化合物的哪些分子描述符，以及这些分子描述符在什么取值或者处于什么取值范围时，能够使化合物对抑制 ERα 具有更好的生物活性，同时具有更好的 ADMET 性质（给定的五个 ADMET 性质中，至少三个性质较好）。

二、问题分析

2.1 问题一的分析

问题一需要针对附件 1 和附件 2 中的 1974 个化合物的 729 个分子描述符进行主要变量筛选，并作为问题而的预测模型变量。主要难点在于如何为保证筛选变量更具有代表性和独立性。首先需要对 1974 个样本数据进行预处理，以降低化合物生物活性过低对整体数据集的影响，提高数据的可用性。其次，对预处理的数据，通过降维的方法，建立综合筛选模型，并进行分析比较，以综合各方法的优势。进一步的，由于各个分子描述符之间存在着很强的相关联性，为保证独立性，需要对改进综合筛选模型得到的变量进行进一步的筛选，利用相关度分析，过滤掉耦合变量的影响。

2.2 问题二的分析

问题二要求依据样本和第一问提取出的分子描述符的变量，通过数据挖掘构建化合物对 Erα 生物活性的定量预测模型，并进行模型验证。

第一问筛选了 1965 组分子描述符样本和 20 个主要变量。神经网络作为一种机器学习算法，具有良好的非线性特征，在函数逼近中应用很广泛。这种算法在多变量回归中优势较大，但适用难以用常规数学模型描述的过程。我们建立了基于遗传算法优化的向后传播神经网络模型（GA-BP）通过已知的变量数据进行对 Erα 生物活性的定量预测。为验证上述分析并比较，同时建立了多元线性回归模型和非线性回归模型，通过三种模型的比较以验证 GA-BP 神经网络预测变量与生物活性关系的有效性与先进性。

2.3 问题三的分析

针对问题三，本题所要求基于吸收，分布，代谢，排泄，毒性五大指标，分别构建分类预测模型以研究预测新型结构化合物在人体内是否具备良好的药代动力学性质及安全性。该题理解为对表格中的 729 个分子描述符，和 1974 个化合物的 ADMET 数据构建分类预测模型，即构建五个结构类似的二分类预测模型。因此，本文讲对三种典型分类预测的集成模型 bagging，stacking，boosting 分和求解，讨论选择出合适的预测算法。

2.4 问题四的分析

问题四是看上去较为混乱，要求从分子描述符中寻找到合适的区间范围，要求该区间范围内分子描述符所影响的样本结果既满足生物活性的要求，又满足 ADMET 性质要求。实际上，第四题是与前三个问题有密切联系的。经过对前三个问题的研究和实现，目前的研究已经对影响生物活性和 ADMET 五大指标影响较大的分子描述符具有一定认识。在本节中所涉及的研究和讨论，应该是基于前文所述的结果之上，提出的模型架构。因此，本文考虑将问题转化为多分类模型的构建问题，采用聚类思想对不同分子描述符内在特性进行挖掘探究。首先将题目所给 1984 条原始数据作为基本研究对象，对其 729 个分子描述变量进行数据降维，随后对于题目所提出的两点约束要求，本文考虑建立一个合理化的特征指标来对化合物能否满足以上两点进行评估。获得药物性分类指

标后，对总体样本进行分类操作，获得多分类问题的目标数据集。最后利用聚类方法，对目标特征变量进行可视化聚类分析，完成特征变量范围的选取。并用评判指标验证其合理性。

三、问题假设

假设 1：附件中的样本中的数据真实，来源可靠，可用于数据挖掘分析，能够作为检验模型准确性的样本；

假设 2：假设附件样本中化合物生物活性和 ADMET 性质仅可能与 729 个分子描述符有关，不再有其他变量的参与。

假设 3：数据样本来自于同一时间段，不考虑由时间、人为因素引起的数据误差。

四、符号说明

序号	符号	含义
1	k	迭代次数
2	r	相关系数
3	Z_i	第 i 个主成分
4	R	相关系数矩阵
5	K	决策树个数
6	W_n	回归系数
7	w	超平面的法向量
8	C	惩罚因子
9	ξ_i	非负松弛变量
10	f_j	染色体的适应度
11	e_{ij}	输入误差
12	β	核的超参数
13	MAE	平均绝对误差
14	MAPE	平均绝对百分比误差
15	RMSE	均方根误差
16	R^2	拟合优度
17	j	表示叶子节点
18	n	交叉验证次数
19	G-Mean	几何平均数
20	w_i	权值

五、问题一(主要变量筛选）

5.1 问题一分析

附件 1 给出 1974 个化合物的活性数据（即样本数据），附件 2 中给出了上述 1974 个化合物的 729 个分子描述符信息（即自变量）。不同的分子描述符，即不同的自变量的变化会对化合物生物活性造成不同程度的影响。本问希望对变量进行重要性排序操作，从 729 个变量中提取出寻找建立生物活性定理预测模型的主要变量，使挑选出的主要变量具有代表性和独立性。另外，题目要求给出前 20 个对生物活性最具有显著影响的分子描述符，并在第二问中将其作为建模变量考虑。

本题的难点在于：（1）各分子描述符信息与化合物生物活性之间相关影响可能不仅是线性相关，也可能非线性相关，判定因、自变量相关程度较为困难，同时为了后续的操作条件优化，选择的变量必须是原有变量，这是特征选取问题，无法使用较为常规的特征提取方法；（2）由于变量过多，变量与变量之间可能存在相互强耦连的关系，故选取变量的独立性问题较难处理。

针对难点（1）——主要变量代表性问题，数据降维算法分为特征选择与降维两大类。特征选择为从给定的特征中直接选择若干重要特征，常用方法的有相关系数矩阵（Pearson\Spearman\Kendall）、距离相关系数和随机森林算法，特征变换为通过某种变换将原始输入空间数据映射到一个新空间中，经典的方法是主成分分析（PCA）。然而，每一种方法都有其优点和缺点，在对复杂的多变量生物活性进行提取主要影响变量的建模中，相关影响变量可能不仅是线性相关，也可能非线性相关，需要有较高的精确度为后续建立模型打下基础。

针对难点（2）——高耦合变量独立性问题，根据得到变量的贡献度排名后，依据从高到低的顺序对变量进行基于斯皮尔曼相关性系数的高相关性滤波，过滤掉耦合变量，以此实现提取具有独立性的特征变量。

基于上述分析，本文实现了四种特征选择和降维方法，并从变量降维过程中采用的算法及处理流程以及变量降维的最终结果两方面对所选择变量的合理性进行评价，综合筛选了主要变量。

整个变量筛选的思路流程图如图 5.1 所示

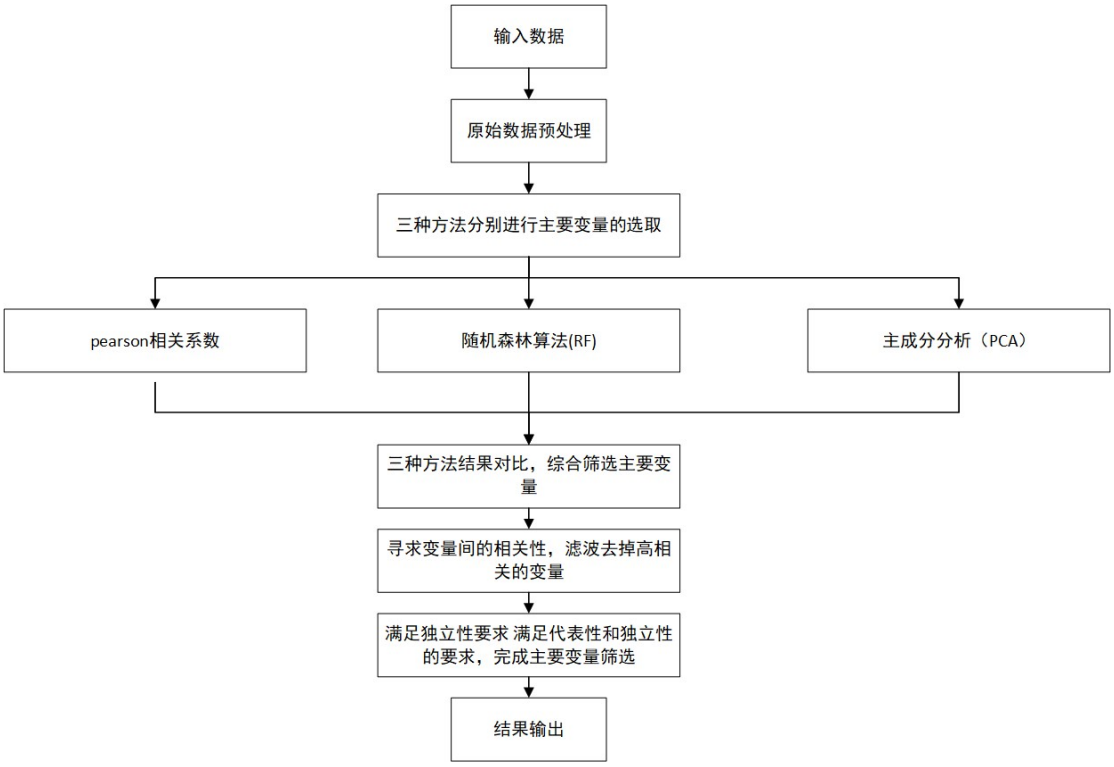


图 5.1 整个主要变量筛选的思路流程图

5.2 问题一求解

5.2.1 数据预处理

在筛选变量之前首先需要对全部 1974 个样本数据进行整定，优化数据的可行性和合理性。

通过对附件 1 的化合物活性数据分析发现，化合物对 ERα 的生物活性值由 IC50 代表，其值越小代表生物活性越大，对抑制 ERα 活性越有效。样本数据的 IC50 范围为 0.046——3500000，之间相差过于悬殊，直接进行变量降维筛选势必会某些偏离值所影响，故需要剔除不合理样本行。从图 5.2 中也可以直观的看出，处理后的数据分布在整体趋势不变的情况下更加集中。

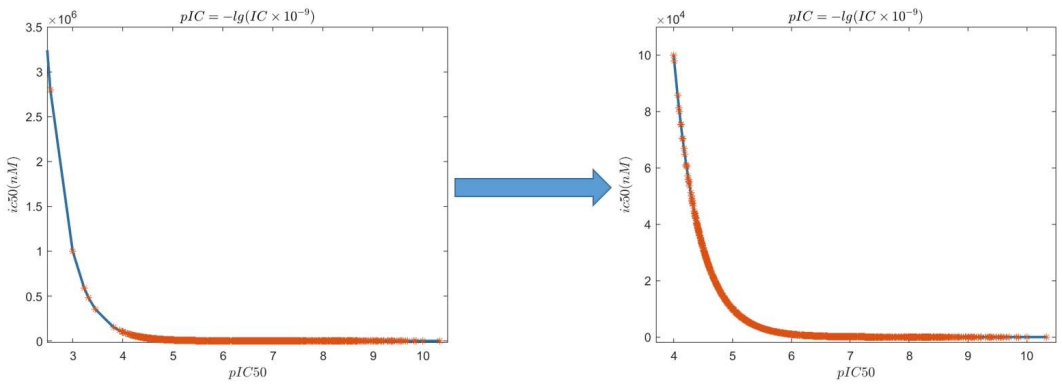


图 5.2 样本数据剔除前后曲线对比

其次，针对附件2中的729个分子描述符信息，我们剔除了全为零值的列，可视为，此分子描述符信息对化合物活性数据不起作用，故不参与下一步的主要变量筛选工作。

表 5.1 数据处理前后数据情况对比

	样本数	分子描述符
处理前	1974	729
处理后	1965	504

5.2.2 相关系数矩阵

相关系数一般用于考察两个事物之间的相关程度。相关系数矩阵法主要有Pearson、Spearman、Kendall三种典型算法。本文主要用到Pearson、Spearman，下面简介如下。

（1）Pearson相关系数

皮尔逊相关系数，又称为皮尔逊积矩相关系数[8]，是用于度量两个变量X和Y之间的相关性，其值介于-1.0与1.0之间。一般用于分析两个连续变量之间的关系，是一种线性相关系数，皮尔森相关系数越大，说明两个变量之间的相关性越高。公式为：其中

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})}}$$

表 5.2 皮尔森相关系数判断范围

系数范围	相关强度
0.0~0.2	极弱相关或无相关
0.2-0.4	弱相关
0.4-0.6	中等程度相关
0.6-0.8	强相关
0.8-1.0	极强相

（2）Spearman相关系数

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2 - 1)}$$

对顺序变量往往使用Spearman相关系数，Spearman相关系数的计算采用的是取值的等级，而不是取值本身。计算Spearman相关系数的公式与计算Pearson相关系数的类似，但用等级代替了各自的取值[10]。

5.2.3 随机森林算法（RF）

随机森林算法一种新兴起的、高度灵活的机器学习算法，常用来评价变量的重要程度。拥有广泛的应用前景，在大量分类以及回归问题中具有极好的准确率。并且，随机森林算法自带特征筛选机制，因此能够评估各个特征在相应问题上的重要性。

随机森林的主要思想是以 K 个决策树为基本分类器，进行集成学习后得到一个组合分类器。从直观角度来解释，每棵决策树都是一个分类器，那么对于一个输入样本，N 棵树会有 N 个分类结果。而随机森林集成了所有的分类投票结果，将投票次数最多的类别指定为最终的输出，这就是一种最简单的 Bagging 思想[11]。

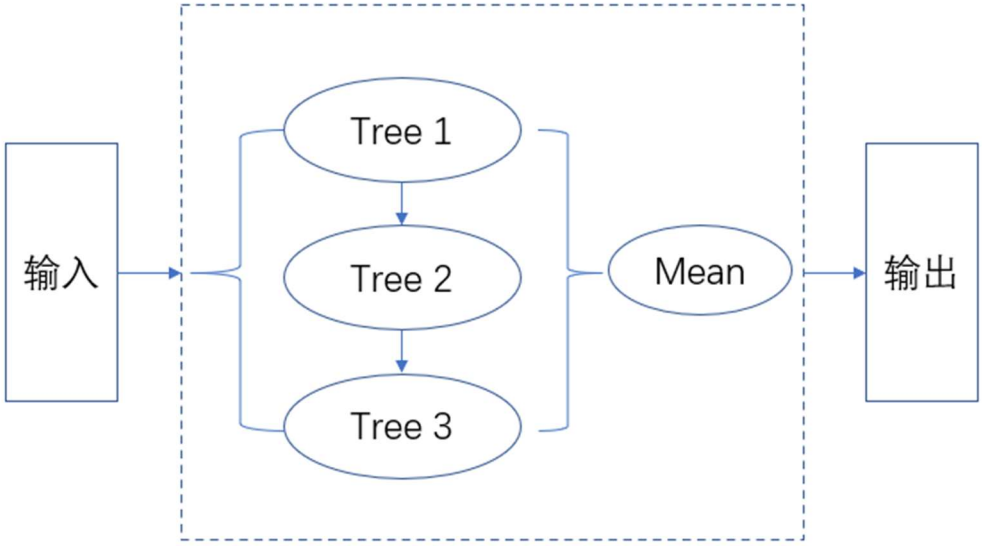


图 5.3 随机森林分类流程

随机森林计算简单，容易实现，计算开销小，对噪声和异常值有较好的容忍性，能够在不需要降维的条件下处理具有高维特征的输入样本，而且能够评估各个特征在分类问题上的重要性，具有良好的可扩展性和并行性[12]。

5.2.4 主成分分析（PCA）

主成分分析是利用降维的思想，在损失很少信息的前提下把多个指标转化为几个综合指标的多元统计 方法。 转化生成的综合指标称之为主成分，其中每个主成分都是原始变量的线性组合，且各个主成分之间互 不相关，这就使得主成分比原始变量具有某些更优越的性能。主成分分析用于对数据信息进行浓缩，比如总共有 20 个指标值，是否可以将此 20 项浓缩成 4 个概括性指标。除此之外，主成分分析可用于权重计算和综合竞争力研究。主成分分析其中权重计算：利用方差解释率值计算各概括性指标的权重；

一个主成分不足以代表原来的 p 个变量，因此需要寻找第二个乃至第三、第四主 成分，第二个主成分不应该再包含第- 一个主成分的信息，统计.上的描述就是让这两个主 成分的协方差为零，几何上就是这两个主成分的方向正交。具体确定各个主成分的方法 如下。

设Z_i表示第i个主成分，$i = 1,2, \cdots, p$，可设

$$\begin{cases} Z_1 = c_{11}X_1 + c_{12}X_2 + \cdots + c_{1p}X_p \\ Z_2 = c_{21}X_1 + c_{22}X_2 + \cdots + c_{2p}X_p \\ \cdots\cdots \\ Z_p = c_{p1}X_1 + c_{p2}X_2 + \cdots + c_{pp}X_p \end{cases}$$

其中对每一个i，均有$C_{i1}^2 + C_{i2}^2 + \cdots + C_{ip}^2 = 1$

具体步骤为：

（1）将原始数据标准化，以消除量纲的影响(软件自动计算)。

（2）建立变量之间的相关系数矩阵 R。

（3）计算相关系数矩阵 R 的特征值λ_m和特征向量u_m。

（4）计算特征值λ_m的信息贡献率和累积贡献率。

5.2.5 三种方法的结果与分析

使用 Matlab 软件对相关系数矩阵法和随机森林算法进行编程实现，使用 SPSS26.0 软件对主成分分析法实现，相关系数矩阵采用相关度图的形式加以展现，主成分分析法采用表格的形式加以展现，主要变量排序用柱状图展现。

首先，对相关系数矩阵中的 Pearson 相关系数编程实现，由于 504 个变量的相关系数矩阵过大，这里只列出按照 Pearson 结果从高相关度到低相关度排列的前 30 个变量的柱状图。

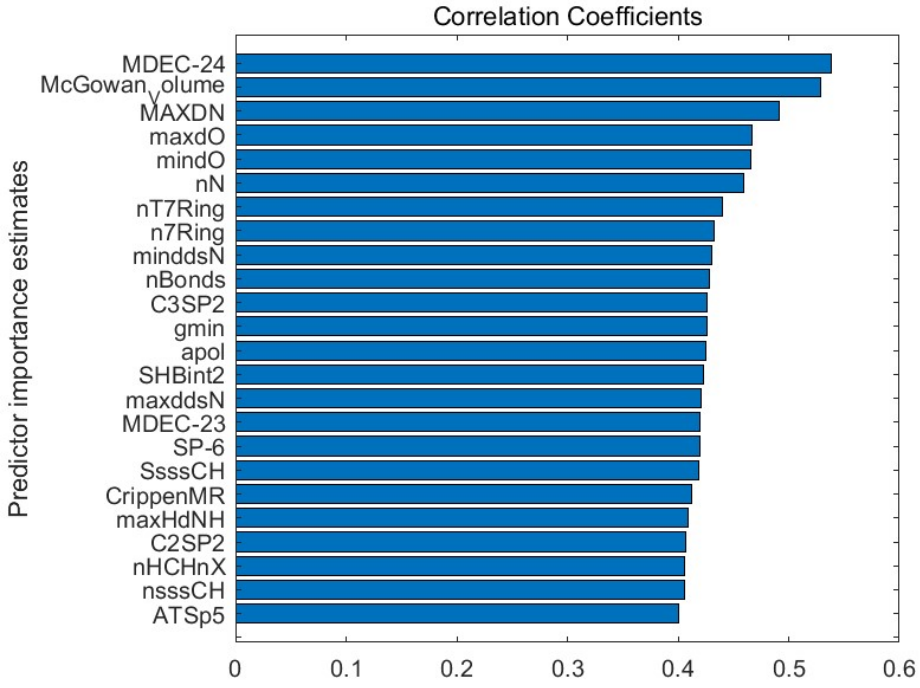


图 5.4 Pearson 相关系数法结果图

对 Pearson 相关系数法结果进行分析，可以发现大部分化学分子符相关性较低，猜测这是因为 Pearson 相关系数表征的是线性关系的变量，而大多数化学分子符和化合物相关性可能包含了非线性关系。

然后，对随机森林算法进行编程实现，综合考虑到算法速度和算法准确率，设定 K =200，运行程序得到 504 个变量的贡献度排名，将贡献度由大到小排序。考虑到下一步的高相关性滤波操作会对进一步变量进行降维，故在这一步中先筛选出排名处于前 30 的变量。

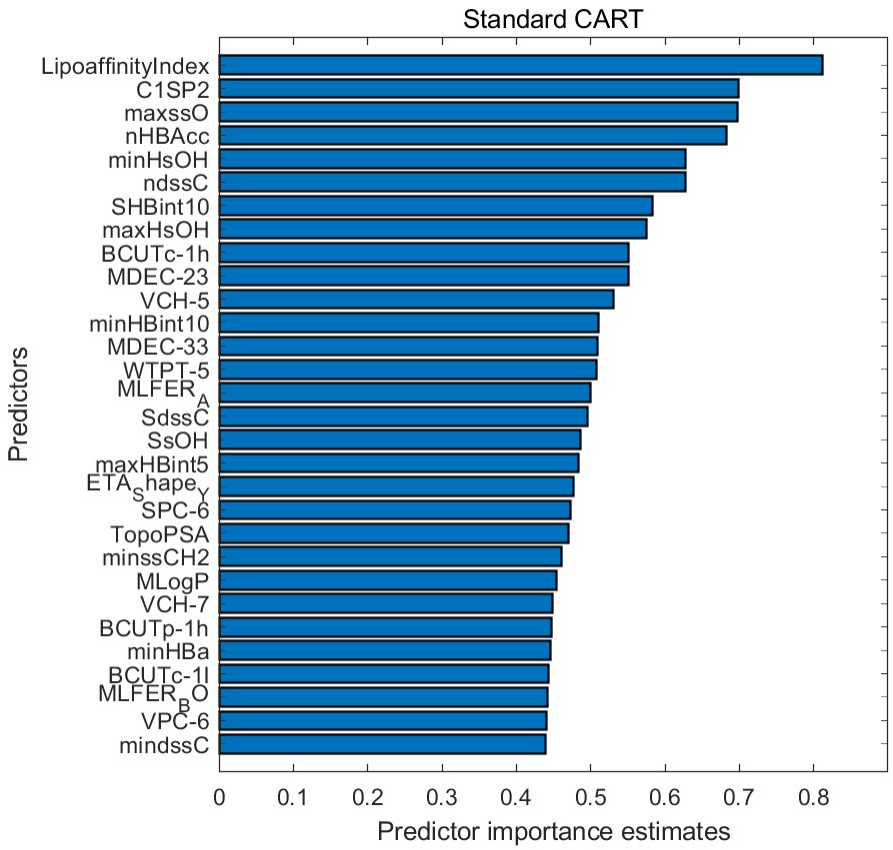


图 5.5 随机森林法结果图

对随机森林法结果进行分析，可以发现重要程度排名前 30 的大部分变量，都超过了 50%，说明选取重要度排名前 30 名的变量已经可以有足够的信息来预测辛烷值。

最后，使用 SPSS26.0 软件对主成分分析法进行分析，部分主成分贡献率见表 1。

表 5.3 主成分贡献率及累积贡献率

成分	贡献率/%	累积贡献率/%
1	19.061	19.061
2	9.722	28.783
3	7.909	36.692
…	…	…
60	0.199	93.967

对主成分分析进行分析，可以发现主成分有 60 个之多，说明需要很多的特征才能完全反映自变量对因变量的贡献，与实际不太相符。主成分分析适用于变量之间存在较强相关性的数据，如果原始数据相关性较弱，应用主成分分析后不能起到很好的降维作用。从程序上也反映了此结论，样本数据在用 SPSS 预处理时数据大部分变量的相关系数都小于 0.3 时，化学分子符和化合物之间的关系较为复杂，应用主成分分析取得的效果不理想。

5.3 主要变量筛选和合理性评价

5.3.1 变量高相关性滤波

经过上述对三种方法所得结果的分析，发现随机森林法更接近实际的题目要求，所筛选的主要变量对化合物生物活性最具有显著影响。而 Pearson 相关系数法和主成分分析得到的结果并不能很好的反映生物活性。因此初步选定随机森林法得到的重要度排名前 30 名的变量作为主要变量候选。

经过上述分析考虑到此题中分子描述符之间具有非线性和高耦合的特征，有必要对其进行高相关性滤波，从高度相关的变量中仅保留重要程度最大的变量，认为它能代表所在高耦合变量组的全部信息。我们选择 Spearman 相关系数距离作为衡量变量间相关性的指标，它采用的是取值的等级，而不是取值本身，比 Person 相关性分析更加适用。

根据 Matlab 程序，对图 5.5 中得到的变量进行高相关性滤波处理，前 30 个重要程度高的变量经滤波处理后剩余 20 个变量，如表 5.4 所示。

表 5.4 对生物活性最具有显著影响的主要变量

排名	符号	符号描述
1	LipoaffinityIndex	脂肪亲和力指数
2	C1SP2	与另一个碳结合的双链碳
3	maxssO	最大的原子型 E-状态：-O-
4	minHsOH	最小原子型 H E-状态:-OH
5	ndssC	原子型 E-状态的统计：=C<
6	SHBint10	路径长度为 10 的潜在氢键的 E-状态的强度统计
7	BCUTc-1h	最高部分电荷加权的 BCUTS
8	MDEC-23	所有二级和三级碳之间的分子边缘距离
9	VCH-5	效价链，顺序 5
10	minHBint10	路径长度为 10 的潜在氢键的最小 E-状态强度
11	MDEC-33	所有三级碳之间的分子边缘距离
12	WTPT-5	从氮原子开始的路径长度总和
13	MLFER_A	总体或总和溶质氢键酸度
14	SdssC	原子类型 E-状态之和：=C<
15	maxHBint5	路径长度的潜在氢键强度的最大 E-状态
16	ETA_Shape_Y	形状指数 Y

17	SPC-6	简单路径簇，顺序 6
18	minssCH2	最小原子型 E-状态：-CH2-
19	VCH-7	价链，顺序 7
20	BCUTp-1h	最高极化率加权的 BCUTS

5.3.2 主要变量筛选合理性评价

（1）从变量选取过程中采用的算法来看：

使用三种不同方法得出结果，并统一分析与优劣，最终采用随机森林得出的重要程度排名的主要变量，最为符合题目要求和逻辑关联。而设计的基于 Spearman 相关性高相关度变量滤波算法则保证了降维后变量之间的独立性，可根据 Spearman 相关系数计算结果对变量之间的相关程度进行可视化，如图 5.6 所示。可见选取的 20 个变量之间相关性低，独立性较好。

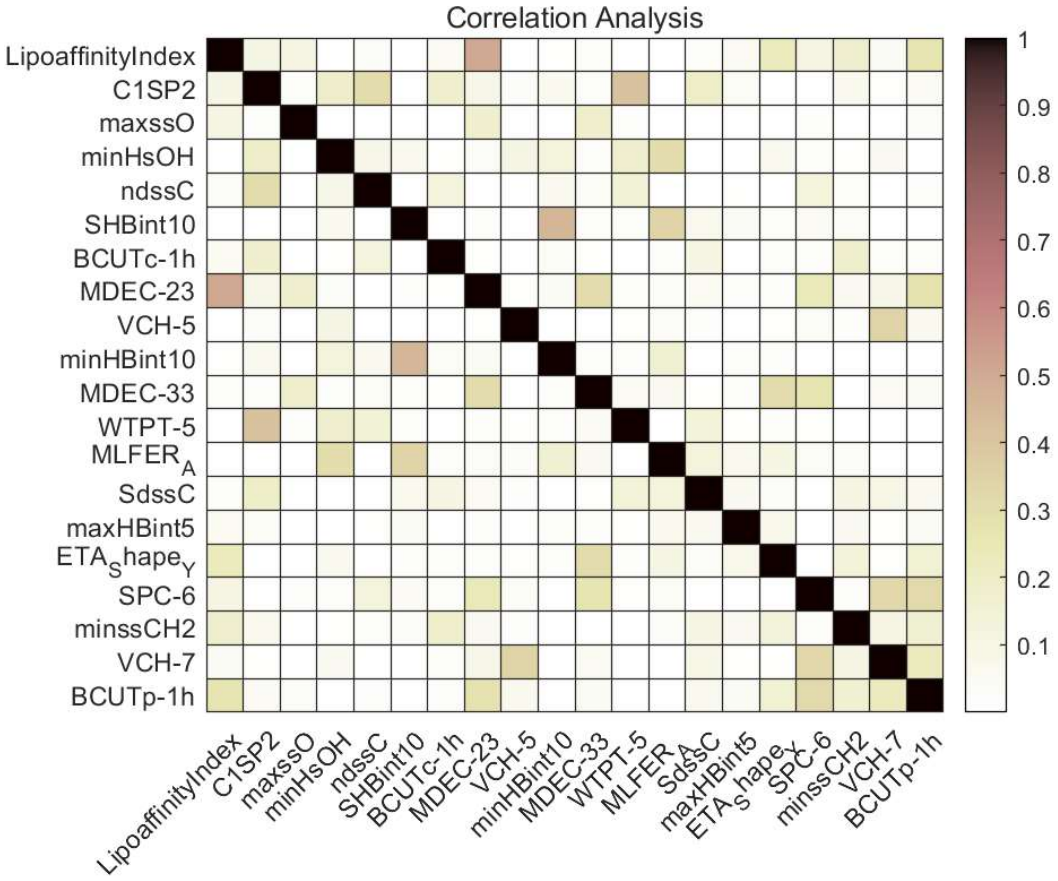


图 5.6 随机森林法结果图

（2）从变量的所属类别可以看出：

从各变量所属化合物分子描述符的统计结果可以看出：所提取变量包括了脂肪亲和力指数、原子型 E-状态、氢键强度统计、分子边缘距离等各种特征的特征，对化合物物理化学性质和拓扑结构特征等反应较为全面。

六、问题二(生物活性的定量预测模型）

6.1 问题二分析

问题二要求依据样本和第一问提取出的分子描述符的变量，通过数据挖掘构建化合物对 ERα 生物活性的定量预测模型，并进行模型验证。

第一问筛选了 1965 组分子描述符样本和 20 个主要变量。神经网络作为一种机器学习算法，具有良好的非线性特征，在函数逼近中应用很广泛。这种算法在多变量回归中优势较大，但适用难以用常规数学模型描述的过程。我们建立了基于遗传算法优化的向后传播神经网络模型（GA-BP）通过已知的变量数据进行对 ERα 生物活性的定量预测。为验证上述分析并比较，本文同时建立了多元线性回归模型和非线性回归模型，通过三种模型的比较以验证 GA-BP 神经网络预测变量与生物活性关系的有效性与先进性。问题二求解的流程图如下所示，根据不同的算法我们在训练集和测试集间优化函数关系，使用最理想的方法对结果做出预测。

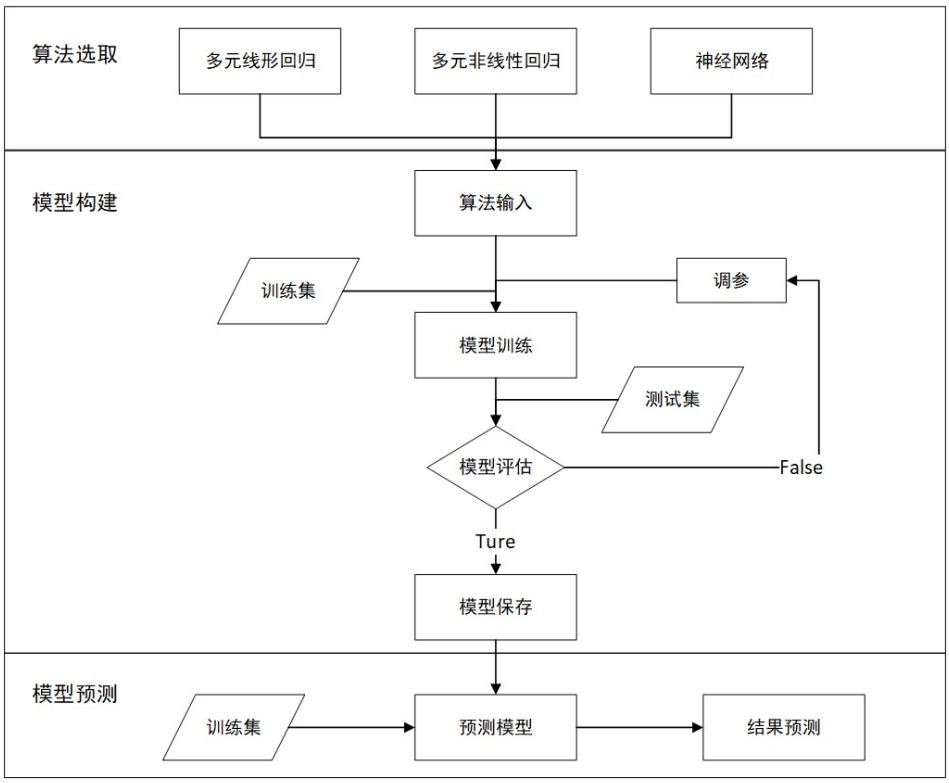


图 6.1 问题二求解流程图

6.2 问题二模型建立

6.2.1 样本数据预处理

通过问题 1 的指标筛选，得到 1965 组分子描述符的 20 个相关变量，即神经网络的输入层，输出层为 1965 组样本对应的 pIC50 的值。前 1600 组为训练集，后 365 组为测试集。基于此对上述数据根据下面的公式进行标准化处理。

$$x^* = \frac{x - x_{min}}{x_{max} - x_{\min}}$$

6.2.2 多元线性回归模型（MLR）

多元线性回归指在实际问题中，一个变量往往受到多个变量的影响。一般形式为：

$$Y_i = b + w_1 x_1 + \cdots w_n x_n$$

其中n 为解释变量的数目，$w_n(n = 1,2,\ldots,k)$称为回归系数,上式也称为总体回归函数的随机表达式，它的非随机表达式为:

$$E(Y|X_{1i}, X_{2i}, \ldots X_{ki}) = w_0 + w_1 x_1 + \cdots w_n x_n$$

多元线性回归模型的参数估计在要求误差平方和为最小的前提下，用最小二乘法求解参数，其均值都为 0，方差为 σ^2 ，即遵从同一正态分布 $N(0, \sigma^2)$ ，这就是多元线性回归的数 学模型。 采用最小二乘法估计总体参数拟合一个带有系数$w = (w_1, \ldots w_p)$的线性模型，使得数据集实际观测数据和预测数据（估计值）之间的残差平方和最小。其数学表达式为:

$$\min_{w} \|Xw - y\|_2^2$$

另外，多元线性回归模型的建立要求变量之间互不相关，即无多重共线性。本文问题一筛选出的 20 个特征变量已经经过了相关性分析，因此满足其要求。

6.2.3 支持向量机回归模型（SVM）

SVM 算法是一种广泛应用于分类及回归问题的机器学习方法，不仅支持线性与非线性的分类，也支持线性与非线性回归。该算法是基于统计学理论、Vapnik-Chervonenkis Dimension（VC 维）理论和结构风险最小化原理的基础上建立而成的，依据有限样本信息在模型中的复杂性和学习能力之间探寻最佳方案，以获得最好的泛化性能。针对非线性不可分问题，则通过核函数将数据由低维空间映射到高维空间，进而实现高维可分。

SVM 的基础是寻找在线性可分条件下的最优分离超平面，首先给定一个样本集 $S = \{(x_i, y_i);\ i = 1, \ldots, n, x \in R^d, y \in \{+1, -1\}\}$其中，$x_i$为数据，$y_i$为数据所属的类别。其分类超平面的表达式及目标函数分别为:

$$f(x) = wx + b$$

$$\min_{w,b}\frac{1}{2}\left||w|\right|^2+C\sum_{i=1}^{n}\xi_i \quad s.t.\begin{matrix}y_i(w\cdot x+b)\geq 1-\xi_i\\ \xi_i\geq 0, i=1,2,\dots,N\end{matrix}$$

其中，w 为超平面的法向量；b 为超平面的平移距离。ξ_i 为非负松弛变量，可以提高模型的泛化能力,C 为惩罚因子，可以用来权衡分类损失和最大间隔之间的关系。

通过 SVM 来进行回归预测的优点如下：

(1)非线性映射是 SVM 方法的理论基础，SVM 利用内积核函数代替向高维空间的非线性映射；

(2)支持向量是 SVM 的训练结果，在 SVM 分类决策中起决定作用的是支持向量。

(3) SVM 是一种有坚实理论基础的新颖的小样本学习方法。它基本上不涉及概率测度及大数定律等，因此不同于现有的统计方法。从本质上看,它避开了从归纳到演绎的传统过程,实现了高效的从训练样本到预报样本的“转导推理”，简化了通常的分类和回归等问题。

(4)有一定的鲁棒性。SVM 模型也有一定的缺点如时间复杂度高、无法对大规模训练样本难以实施等。但本文的训练样本小于 2000 个，所以综上 SVM 模型能完成生物活性与 20 个变量之间非线性关系的预测。

本文选用高斯 RBF（Radical Base Function）函数作为核函数，高斯 RBF 为空间中任一点 x_i 到某一中心 x_j之间欧氏距离的单调函数，具有较好的效果，表达式为：

$$k\left(x_i,x_j\right)=e^{-\frac{\left(x_i,x_j\right)^2}{\sigma^2}}$$

其中，核函数参数σ影响着从样本空间到特征空间的映射。C 和σ对于 SVM 有很大的影响，所以合适模型参数的选取至关重要。

6.2.4 高斯过程回归模型(GPR)

高斯过程回归具有结构简单、参数少、预测输出具有概率特性等特点，在处理非线性、高复杂度的数据上有较好应用。它通过处理训练数据的变化找到规律，并且估计生成先验分布，根据数据先验分布结合贝叶斯原理完成对后验分布的计算。因此高斯过程回归也可视作一种非线性概率模型，通过选用不同

的核函数来无限逼近真实数据，在解决实际问题时，输出均值的同时给出置信区间，使结果的有效性不断增强。

在给定数据的有限集合中，$f(x)=x^Tw$ (其中x为输入向量，w为线性模型参数的权向量)可构成随机变量的一个集合，且具有联合高斯分布，高斯过程的统计特征由其均值函数$m(x)$ 和协方差函数 $k(x,x')$ 组成，因此有

$$f(x)\sim GP\big(m(x),k(x,x')\big)$$

将含噪声考虑到观测目标值y中，可建立高斯过程回归问题的一般模型，即

$$y=f(x)+\epsilon$$

式中：ϵ为独立的高斯白噪声，符合高斯分布，均值为 0，方差为σ^2，记做 $\epsilon\sim N(0,\sigma^2)$ 根据贝叶斯定律以及自变量 x通过核函数从低维到高纬的映射，我们可以得到观测值 y和预测值f_*的联合先验分布：

$$\begin{bmatrix}y\\f_*\end{bmatrix}\sim N\left(0,\begin{bmatrix}K(X,X)+\sigma_n^2I & K(X,X_*)\\ K(X_*,X) & K(X_*,X_*)\end{bmatrix}\right)$$

其中，$K(X,X)$为训练点的协方差矩阵，$K(X_*,X_*)$为测试点协方差矩阵，$K(X,X_*)=K(X_*,X)$为训练点和测试点之间的协方差矩阵，并由此得到主要的 GP 回归方程，即：

$$f_*\mid X,y,x_*\sim N(\overline{f_*},cov(f_*))$$

其中，预测均值向量$\overline{f_*}$就是 GP 回归模型的输出，也即输出向量f_*的预测值。由上述公式可以看出，高斯过程回归算法，对于预测输出值，并非单纯估算一个预估值，还会估算出预估值方差。这种方式将预测的不确定性考虑进去，更贴近实际。

GPR 模型的优点之一是：基于训练数据的训练过程就是对超参数的选择过程。例外：它使得在参数范围内进行计算变得容易进行，并且这种模型可以进行灵活的调整。预测值的均值是核函数 K(x^i,x^j)的线性组合，可将非线性关系的数据映射到特征空间后转为线性关系，使复杂非线性问题转化为线性问题。高斯过程可选择不同的协方差函数，本文采用最常使用的有理二次协方差函数（Rational Quadratic）作为核函数来进行模型的训练与预测。

6.2.5 神经网络模型

简单 BP 神经网络模型

BP 神经网络，是一种基于误差反向传播算法的前馈型神经网络，其结构相对较为简单。它模仿人脑神经元对外部信号的应激过程，利用信号正向传播和误差反向调节的学习机制，经多次迭代，最后得到处理非线性信息的网络模型。其网络结构拓扑图如下图所示，以输入层 3 个特征变量，隐藏层 4 个神经元节点，单目标输出为例，权值和阈值则作用于在输入层与隐含层以及隐藏层与输出层的连接。

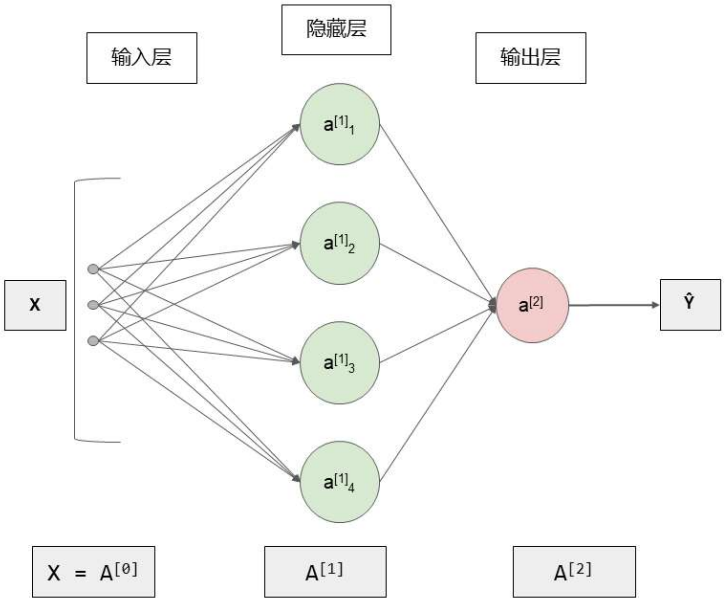


图 6.2 神经网络结构拓扑图

建立的简单 BP 神经网络模型的主要参数设置如下: 输入层节点数为 20，输出层节点数为 1，单隐藏层，隐藏层神经元节点数为 10，激活函数 σ =tanh(·)，其最大训练次数设为 1000，收敛目标设置为 0. 001，学习率为 0. 01，其余参数设为默认值。

神经网络遗传算法优化模型

遗传算法是根据生物染色体遗传时发生选择、交叉、变异的原理，对解集进行更新优化。基于遗传算法优化的 BP 神经网络模型如图所示，其由神经网络结构的确定，遗传算法优化和 BP 神经网络预测 3 个部分组成。神经网络结构确定部分是根据拟合函数输入输出参数个数确定神经网络结构，进而确定遗传算法个体的长度。遗传算法优化，使用遗传算法优化 BP 神经网络的初始权值和偏置，其基本思想就是用个体代表网络的初始权值和偏置、个体值初始化的 BP 神经网络的预测误差作为该个体的适应度值，通过选择、交叉、变异操作寻找最优个体，即最优的 BP 神经网络初始权值。

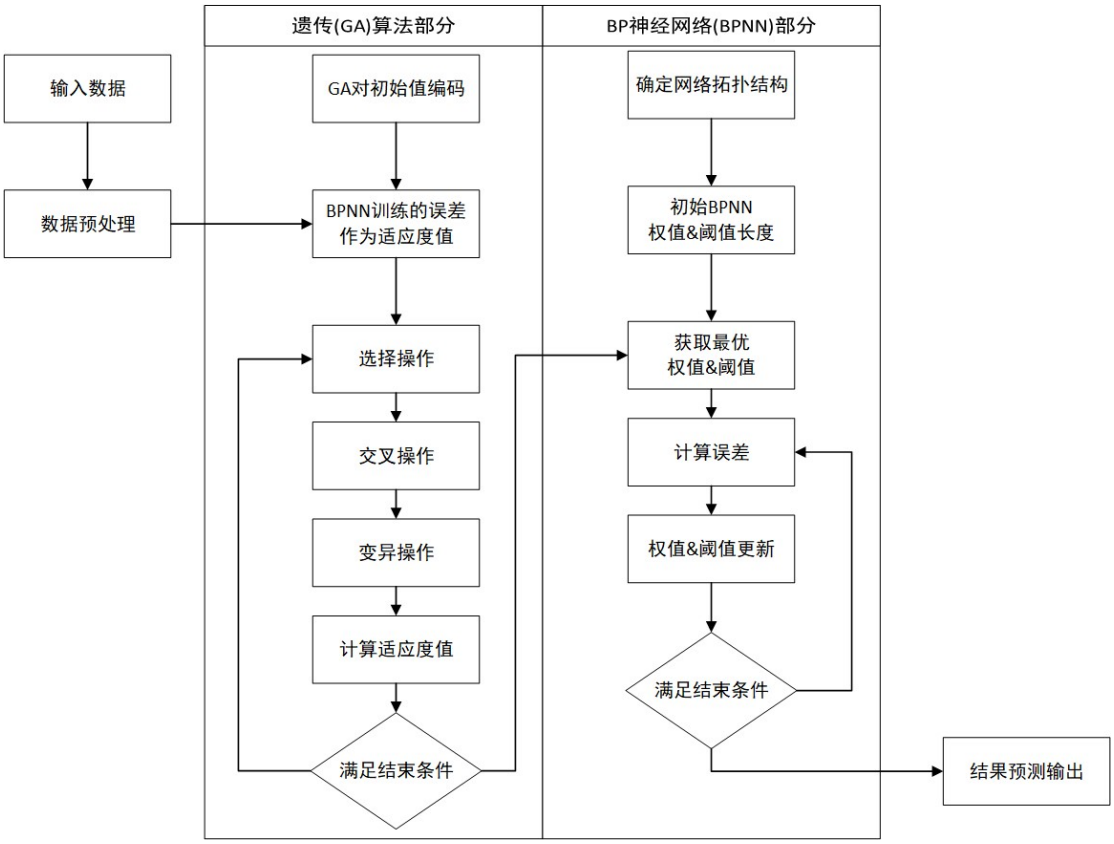


图 6.3 遗传算法—神经网络算法示意图

在算法中, 每个染色体代表待处理问题的个体, 并且许多染色体(基本单元)形成遗传算法的初始种群, 通过选择、交叉、变异三个步骤的操作, 将神经网络从前一代(即父母)群体进化到下一代来完成优化过程. 在本文中, 遗传算法中群体的适应度基于 GA-BPNN 与编码染色体的预测误差，每个染色体的适应度值(f_j)通过下式计算:

$$f_j = \left[\frac{1}{n}\sum_{j=1}^{m}\sum_{i=1}^{n} e_{ij}^2\right]^{-\frac{1}{2}}$$

个体被选择的概率(P_j)由下式计算：

$$P_j = f_j / \sum_{j=1}^{m} f_j$$

其中，f_j是第j个染色体的适应度值，n是神经网络输入数据集的数量，m是遗传算法中种群的大小，e_{ij}是第i个神经网络的输入与第j个染色体间的误差.

接着交叉, 将相互配对的父母染色体依据概率交换部分信息，形成下一代。在交叉的过程中交换的部分是在染色体上随机选取。当交叉过程完成后，重新计算每个子代染色体的适应度值，将适应度值高的子代保留在当前种群中。在将子代染色体重新插入原始群体之前, 进行变异过程用来提高遗传算法的搜索能力和群体多样性. 变异是根据变异概率将染色体中的部分编码基因改变来实现. 通过该过程, 遗传算法可以更好地搜索整个参数空间, 并且可以避免陷入局部最优。

本文遗传算法的编码方式采用精度高的实数编码，其便于大空间搜索编码长度由公式 $L = m \times n + n \times l + n + l$确定.其中$m$为输入层神经元个数, n为隐藏层神经元个数, l为输出层神经元个数。完成编码过程后，选择合适的初始化种群以及适应度函数计算公式，进行遗传算法的选择、交叉、变异的迭代过程，当迭代过程满足训练目标要求或迭代次数达到设定的目标时停止迭代。将最优的染色体(即神经网络的初始权值和偏置)作为结果返回。

之后将遗传算法的返回结果作为 BP 神经网络的初始权值和偏置，完成神经网络学习过程，得到的最优神经网络结构即为 GA-BP 神经网络生物活性的定量预测模型。隐含层神经元个数根据经验取为 15。隐藏层和输出层的激活函数分别为 Tanh 和 Sigmod 函数。由于 BP 神经网络的偏置和初始权值采用随机生成的方式，导致 BP 神经网络存在收敛速度慢、不能确保收敛到全局最优值等问题。本杰采用的遗传算法优化 BP 神经网络的初始权值和偏置，通过处理编码变量字符串(即染色体)的聚合，可以加速网络的收敛速度，提升预测模型精度。

6.3 问题二结果分析对比

6.3.1 模型结果

多元回归模型的求解

首先，通过 Matlab 将 1965 个样本随机抽取了 1600 个样本。然后利用这 1600 个样本的指标值求解线性方程，得到的前 20 个分子描述符相关变量对生物活性指数 pIC50 的线性回归方程如下：

$$y_{pred} = -0.2 + 0.145x_1 - 0.218x_2 - 0.087x_3 + \cdots + 3.743x_9 + \cdots + 3.552x_{16} + \cdots 0.324x_{20}$$

其中 x_i (i=1,2,⋯20)为第一问中 20 个变量的顺序排序。

我们把测试集中的数据代入上述方程得到的，预测结果与真实值之间的均方根误差为 0.934，平均绝对误差 0.75，平均绝对百分误差为 17.24%。由此可知多元线性回归模型基本不适用于求解本问题。

支持向量机模型的求解

对于数据集的处理与上述线性回归方法相同，也将 1965 个样本打乱并随机

抽取 1700 个样本。然后利用这 1700 个样本进行训练，剩下的 265 个作为测试集数据，这里我们采用 10 折交叉验证来重复运用随机产生的子样本进行训练和验证。模型其他参数的选择为核函数参数 σ =0.258，指定松弛变量 ξ =0.026 固定。得到的前 20 个分子描述符相关变量对生物活性指数 pIC50 的 SVM 回归预测值与真实值的响应图如下所示，模型运行时间约 3.7 秒，通过计算，得到了预测结果与真实值之间反归一化前的均方根误差为 0.115，平均绝对误差为 0.08，平均绝对百分误差为 7.71%。通过分析预测结果的评价指标我们可以发现，支持向量机对于样本量较小的数据集预测精度较好，而且预测时间短，很适合用于通过分子描述符来预测生物活性指标。

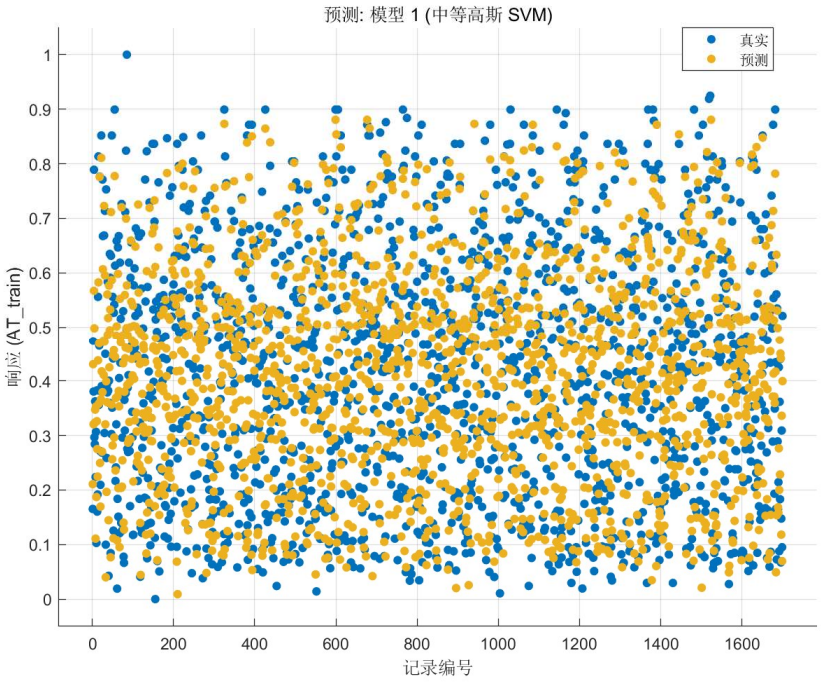


图 6.4 pIC50 的 SVM 回归预测值与真实值的响应图

高斯回归模型的求解

模型训练的方法和支持向量机相同最大似然估计，在训练的过程中，主要需要估计的系数有三个：β，σ^2和θ（核的超参数）即：

$$\hat{\beta}, \hat{\sigma}^2, \hat{\theta} = \arg\max \log P(y|X, \beta, \theta, \sigma^2)$$

对基于高斯过程的回归建模预测，我们只需要确定初始超参数即可，算法会在高斯过程的训练中，使用迭代法来获取最优超参数，在难以得到良好的自整定参数情况下显示出巨大的优越性，在不损失性能的条件下容易实现。同时，高斯过程是一种有着概率意义的核学习机，克服了支持向量机估计输出的缺点。如图，我们利用了高斯过程回归对 pIC50 进行了预测，数据集划分与上述相同

并采用 10 折交叉验证，初始超参数设为 β =0.258, σ =0.026, θ =0.026 运行时间大概为 89.4s，通过计算，得到了预测结果与真实值反归一化前的均方根误差为 0.109，平均绝对误差为 0.081，平均绝对百分误差为 7.21%。通过分析预测结果的评价指标我们可以发现，高斯过程回归对于样本量较小的数据集预测精度非常好，预测误差分布均匀，但预测所花时间跟其他算法相比消耗更多，用其通过分子描述符来预测生物活性的效果好。

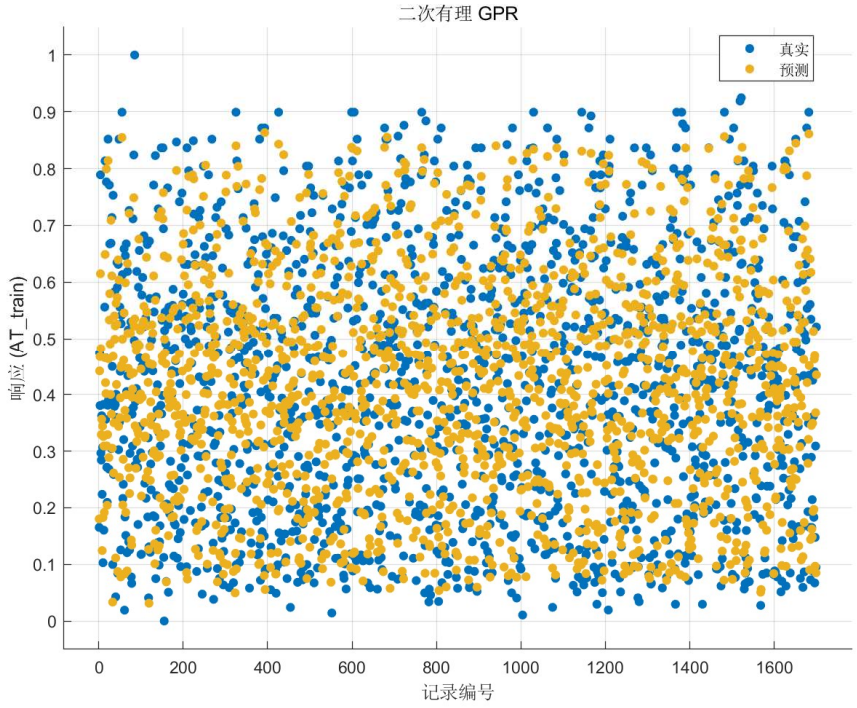


图 6.5 pIC50 的 GPR 回归预测值与真实值的响应图

神经网络遗传算法优化模型的求解

图 6.6 是我们通过两种神经网络得到的拟合度图，可以看出经过遗传算法的优化 BP 神经网络的拟合度有了进一步的提升。BP 网络的运行时间接近都在 15 秒上下，但第一次遗传优化需要消耗大量的时间约为 19 分钟。两种网络预测的均方根误差分别为 0.68 和 0.64，平均百分比误差为 11.85%和 7.51%。通过分析预测结果的评价指标，我们发现 BP 神经网络在预测任务尤其是其经过优化后能预测得到良好的结果。

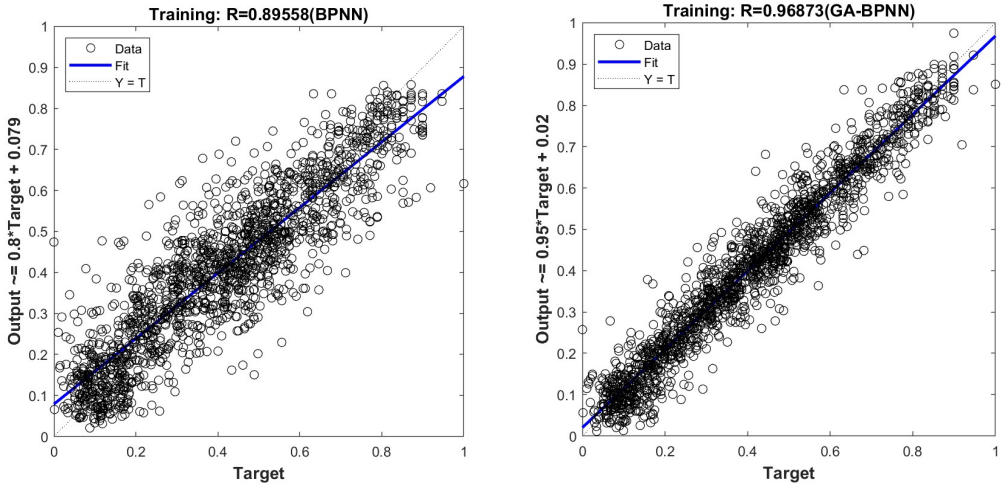


图 6.6 神经网络和经过遗传算法优化的回归拟合度图

以上是我们所选择的各种模型及其实验结果，通过预测结果图能够看出基本上都能够对产品辛烷值变化规律进行拟合，但是具体的误差和误差分布情况我们需要进一步分析，在下节中，我们通过给出模型评判方法，选择的不同评价指标综合对模型预测误差进行分析。

6.3.2 模型评价

前人通过对模型的选择、模型建立和实验，得到了各种模型的评价指标，下面我们通过可视化的方式对所选择的各种模型进行分析。模型评价方法：本研究采用相关系数平均绝对误差 MAE、平均绝对百分比误差 MAPE、均方根误差 RMSE、拟合优度 R^2 4 个指标对模型进行评估,各指标计算方法如下:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_{pred} - y_{exact}\right|$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_{pred} - y_{exact}\right|$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_{pred} - y_{exact}\right)^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_{pred} - y_{exact}\right)^2}$$

$$R^2 = 1 - \frac{\sum(y_{pred} - y_{exact})^2}{\sum(y_{exact} - \bar{y}_{exact})^2}$$

结合 6.3 中的各模型计算结果，我们可以得到五种模型的不同评价指标，如表 6.1 所示，并画出不同算法评价指标直方图 6.7 来直观展示各算法间差异

表 6.1 五种模型的不同评价指标

指标 / 方法	R^2	MAE	$MAPE$	$RMSE$
多元线性回归	0.53	0.746	17.24%	0.934
高斯过程回归	0.77	0.485	7.21%	0.689
支持向量机回归	0.73	0.493	7.71%	0.693
BP 神经网络	0.736	0.504	11.85%	0.683
遗传优化的 BP 神经网络	0.795	0.467	7.51%	0.641

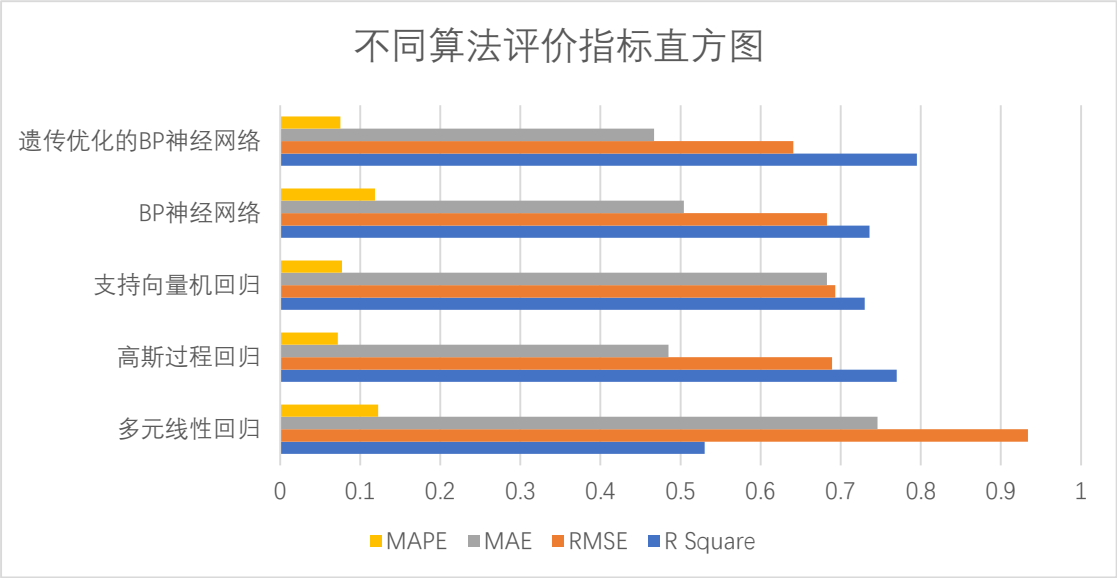


图 6.7 不同算法评价指标直方图

通过对不同算法评价指标的分析，可以发现，对于拟合优度（R 方），高斯过程回归模型和遗传优化的神经网络靠近于 1，表示高斯过程回归模型和 GA-BP 神经网络的回归直线对观测值的拟合程度良好；对于均方根误差，非线性多元回归模型和神经网络较低，表示非线性多元回归模型和神经网络误差分布均

匀，相较于线性回归算法误差波动较为平稳；对于平均绝对误差，神经网络和非线性过程回归模型较低，在遗传算法优化后，BP 网络可以降到更低，这就意味着 BP 神经网络能够很好的预测生物活性的变化，预测结果于真实值之间的误差相对于其他算法要低；对于平均绝对百分比误差，GPR,SVM 和 GA-BP 神经网络相对其他算法都降低了，表示这三种模型所预测的结果中每个样本的平均误差均低于其他两种模型。进一步表明了高斯过程回归和 GA-BP 神经网络在本题所述任务中的优越性。 最后可以看出在经过遗传算法对 BP 网络初始参数的优化后，所有评价指标都得到了提升。

但对于 BP 神经网络，虽然效果较好，无论是优化前还是优化后这种模型容易陷入局部最优，而且模型收敛速度很慢、导致训练时间比较长。而且模型结构只能通过经验确定，获得以上实验结果需要多次模型参数调试。

最后我们通过训练得到的 GA-BP 网络和文件“Molecular_Descriptor.xlsx”中表 test 中的数据完成了对相应分子描述符的生物活性 pIC_{50} 的预测，并依据 pIC_{50} 与 IC_{50} 之间的关系式 pIC_{50}=-lg (IC_{50}×10^(-9))对 PIC50 进行补完。由于篇幅原因，预测见附录 1。

为了直观展示预测效果，我们样本数为横轴，化合物生物活性 pIC_{50} 的预测值为纵轴画出了预测效果图。其中不同颜色的点代表其活性值所在的区间。

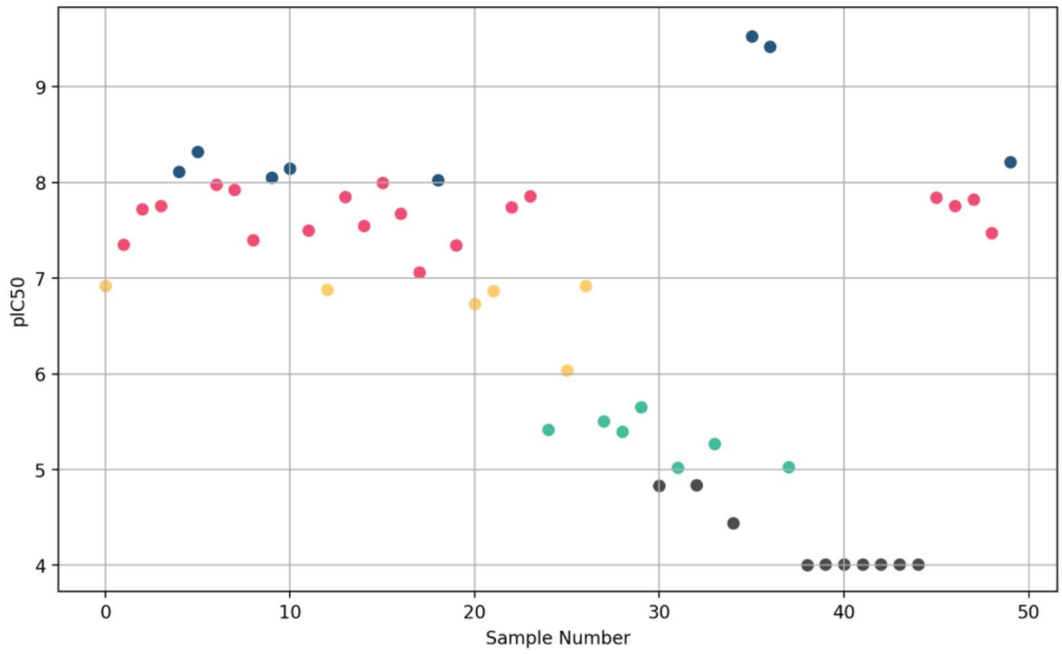


图 6.8 生物活性 pIC_{50} 预测效果图

七、问题三(ADMET 分类预测模型）

7.1 问题三分析

针对问题三，本题所要求基于吸收，分布，代谢，排泄，毒性五大指标，分别构建分类预测模型以研究预测新型结构化合物在人体内是否具备良好的药代动力学性质及安全性。该题理解为对表格中的 729 个分子描述符，和 1974 个化合物的 ADMET 数据构建分类预测模型，即构建五个结构类似的二分类预测模型。因此，本文首先通过文献阅读，从三种典型分类预测的集成模型 bagging，stacking，boosting 中，讨论选择出合适的预测算法。

在找到合适的模型算法后，准备模型所需的数据集并对其进行划分。采用原始数据“Molecular_Descriptor.xlsx”中，表一“training”数据打乱排序后进行分割。考虑选择该数据集中的 80%数据作为训练数据进行训练；20%数据作为测试数据用以验证模型的可靠性。之后训练数据，建立分类模型，根据交叉验证对其进行调参优化。保存已经构建好的模型后，打印模型预测指标，对不同模型的预测指标其进行可视化比较，最后选择出最佳预测模型。在此问题中，可能用到的开源第三方库模块包括：pandas，numpy 读取数据；sklearn 划分数据结构；xgboost 构建模型；matplotlib，plotnine 后处理。综上所述，本文对问题三考虑的思维框架如下图 7.1 所示。

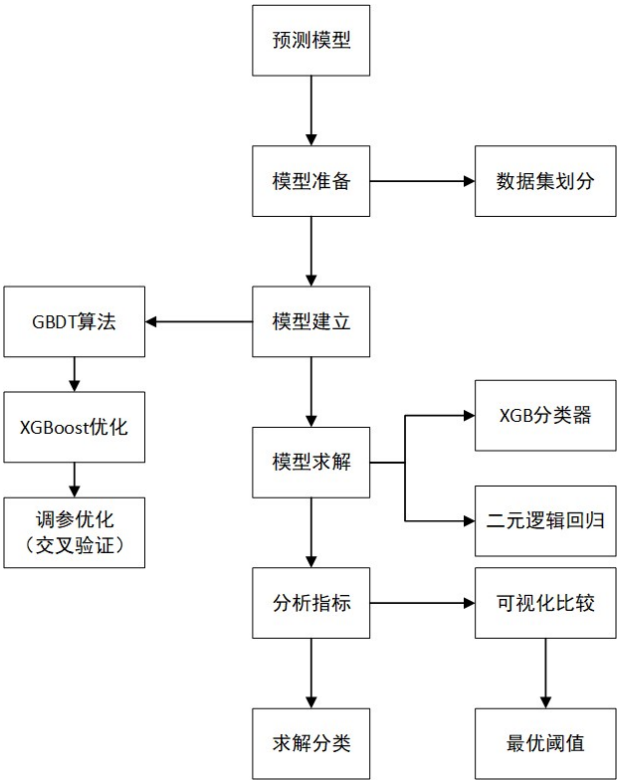


图 7. 1 分类预测模型思维框架

7.2 ADMET 分类预测模型的建立

7.2.1 GBDT 算法原理

梯度提升决策树（Gradient Boosting Decision Tree，GBDT）是一种集成 boosting 思想（Boosting Tree Frame）的算法模型，其本质上属于 GB 框架下的加法模型，由多种弱学习器组成。GBDT 模型可解释强，实施效果好，被广泛应用于数据挖掘等领域[6]。
有学者的研究报告说明，对于样本量较少、变量多变量之间强耦合的场景，GB 模型比支持向量回归、随机森林和多层感知器模型提供了更好的预测精度[7]。

作为集成模型框架下的算法代表，GBDT 采用的多模型策略，本身由多个弱学习器组合而成，这样的组合策略能够很好并有效的解决降低模型偏差的问题。在使用 GBDT 算法模型训练时，采用前向分布算法进行贪婪的学习，即每次进行迭代都学习一棵 CART 树来拟合之前 (t-1) 棵树的预测结果与训练样本真实值的残差。

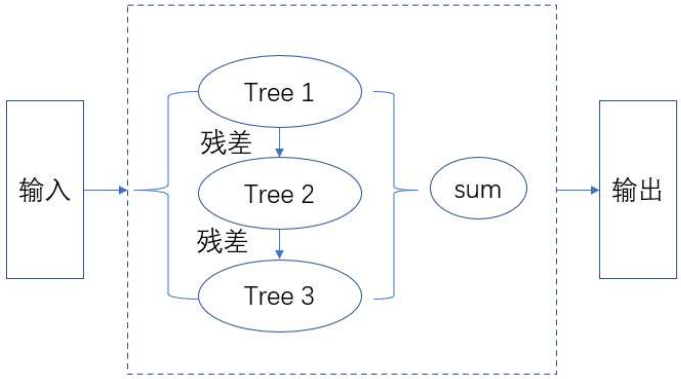


图 7.2 决策树逻辑图

7.2.2 XGBoost 优化

XGBoost（eXtreme Gradient Boosting）极致梯度提升，最早由华盛顿大学计算机科学陈天奇博士在一篇论文中正式提出[9]。。XGBoost 本质上是基于 GBDT 的底层的优化算法，其具有高效、灵活和轻便的特点，在数据挖掘等领域得到广泛的应用。其本身聚焦于对 GBDT 目标函数和求解最优解上的性能优化，对项目工程的计算分析有很大的帮助。XGBoost 对 GBDT 算法的优化可概括为，利用算法优化 GBDT 算法的原始目标函数，得到优化后的目标函数和决策叶子节点上的目标函数构成矩阵，问题转化为求解该矩阵的最优值。

除了速度优化上，XGBoost 其本身就具备有 Boosting Tree 结构的特有优势，例如，无需归一化（Normalization）处理。Boosting Tree 模型通过遍历特征所有取值来选择划分点，是否归一化并不影响这个过程的进行[13]。

7.2.3 模型建立

(1)目标函数

在本文中，以训练集$T=(x1,y1),(x2,y2)\ldots(xn,yn)$，损失函数$l(y_i,\hat{y_i})$，正则化项$\Omega(f_k)$为例，
整体的目标函数可记为，

$$L(\Phi)=\sum_i l\left(y_i,\hat{y_i}\right)+\sum_k \Omega\left(f_k\right)$$

三个步骤优化目标函数:
第一步：二阶泰勒展开，去除常数项，优化损失函数项；
第二步：正则化项展开，去除常数项，优化正则化项；
第三步：合并一次项系数、二次项系数，得到最终目标函数。
最终通过数学变化，得到本文中 XGBoost 的最终进行计算的优化目标函数，表示为，

$$L^{(t)}=\sum_{j=1}^{T}[\,G_j w_j+\frac{1}{2}(H_j+\lambda)w_j^2]+\gamma T$$

其中，T为叶子节点的总数，j 表示叶子节点，G_j表示所有叶子节点上包含样本的一阶偏导数之和，H_j 表示所有叶子节点上包含样本的二阶偏导数之和，w 表示叶子节点的权重向量。

值得注意的是，经过变换后的目标函数变量只剩下了叶子节点的权重向量w，这里可以将单个叶子节点（第j个）的目标函数（权重向量作为变量），写为一元二次方程的形式，

$$f(w_j)=G_j w_j+\frac{1}{2}(H_j+\lambda)w_j^2$$

问题本身被转化为求解$T\times 2$ 阶矩阵，此时达到最优解目标值可以写为,

$$Obj=-\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda}+\gamma T$$

(2)分裂查找算法

当构建决策树时，最关心的问题在于树在何时能分裂。本文思考采用精确贪心算法(Basic Exact Greedy Algorithm)，即在所有特征上，枚举所有可能的划分。精确贪心算法从树的根节点开始，对每个叶节点枚举所有的可用特征，本质上属于一种暴力算法。为了更高效，该算法必须首先根据特征值对数据进行排序，以有序的方式访问数据来枚举打分公式中的结构得分（structure score）的梯度统计(gradient statistics) [13]。简而言之，我们通过两层循环穷举结构得分和梯度统计这两个参数，然后逐个尝试后保留最优的切分方案。

确定完成分裂算法后，开始对每个树进行切分，选取信息增益作为切分准则，信息增益可以定义为

$$Gain = \frac{1}{2}[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} + \frac{G_L^2+G_R^2}{H_L+H_R+\lambda}] - \gamma$$

通过计算每个节点分裂前后的信息增益，选择出信息增益最大的进行切分算法操作，最终可实现本文中决策树的构建。图 7.3 描绘的即为本文所构建的决策树，特征值即为本题所研究的分子描述符变量。

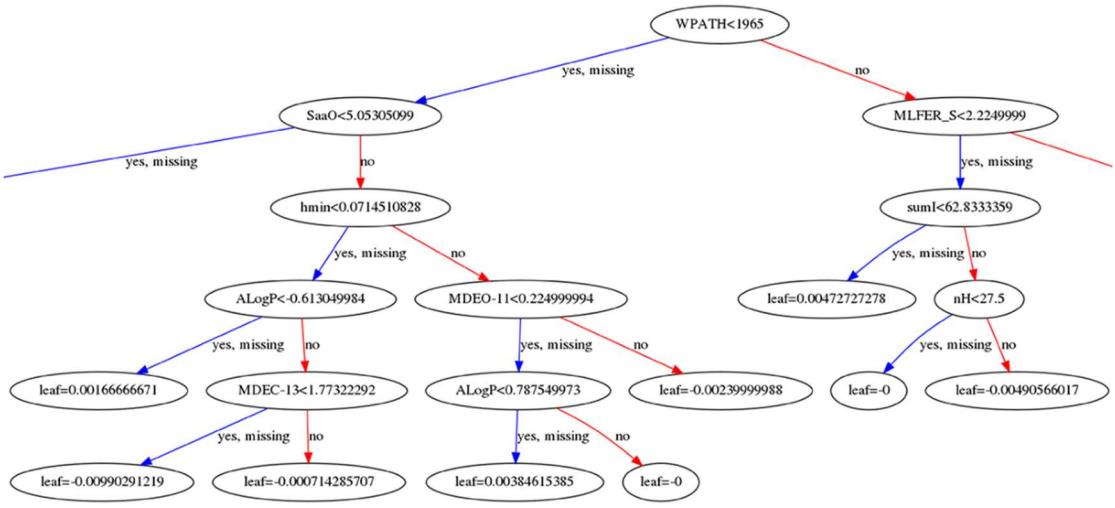


图 7.3 本节构建的决策树（部分展示 ）

(3)程序架构

本题所采用的 Python 代码基于 Xgboost 框架开发，其框架逻辑的 UML 活动图由图 7.4 进行描述。

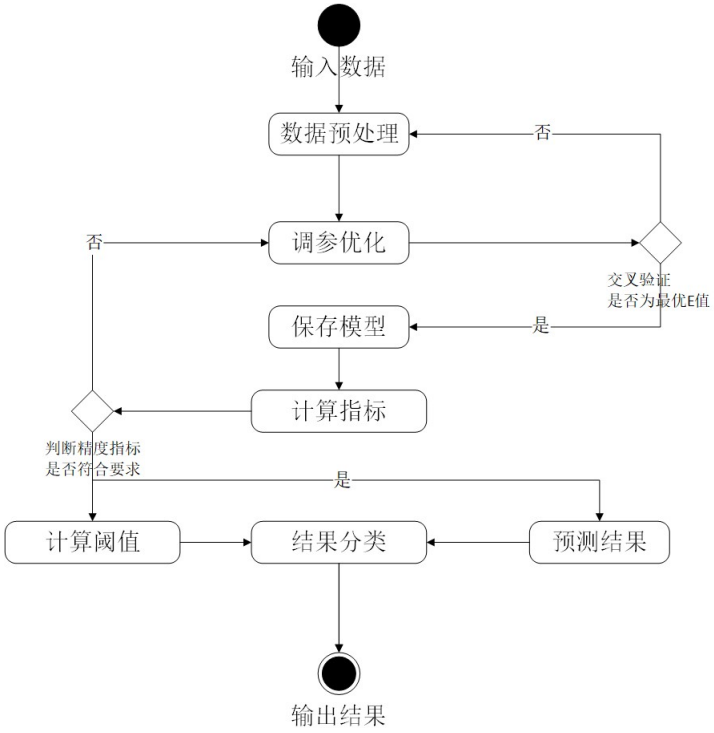


图 7.4 UML 活动图（问题三）

7.3 对比优化

7.3.1 主要特征指标

本题所要求基于吸收，分布，代谢，排泄，毒性五大指标，分别构建 5 套分类预测模型。由于篇幅限制，本文着重对第一套吸收（Absorption）指标的分类预测进行对比分析和讨论。

并非一种分类指标可以评价所有的模型的优劣，因此需要引入对各种类型分类指标的对比优化为了更好的分析和评价模型预测效果，这里引入了几种典型的特征指标作为评价标准，其中包括，ROC 曲线（ROC curve），F1 度量（F1 Score），AUC，准确度（Accuracy），召回率（Recall），特异性。

- **准确度**-正确预测的样本与总样本的比率
- **召回率**或**敏感度**-正确预测的正例样本与实际上所有正例样本的比率
- **特异性**-正确预测的反例样本与实际类别中所有反例样本的比率
- **精确度**-正确预测的正例样本与总预测正例样本的比率
- **F1 分数**-精确度和召回率的调和平均数。因此，这个分数同时考虑了假正例和假反例

- **AUC**-指标用于评价分类器对于正、负样例的辨别能力，对出结果的排序位置的敏感。

7.3.2 调参优化

由于本文所选用数据集的数据量并不大，属于中小型数据集，因此考虑采用交叉验证的方法来评估模型的预测性能，以减小过拟合实现调参优化。交叉验证是典型的观察模型稳定性的方法，其本质是将数据划分成 n 份，依次使用其中一份作为测试集，其他 n-1 份作为训练集，多次计算模型的精确性来评估模型的平均准确程度。训练集和测试集的划分会干扰模型的结果，因此用交叉验证 n 次的结果求出的平均值，是对模型效果的一个更好的度量。

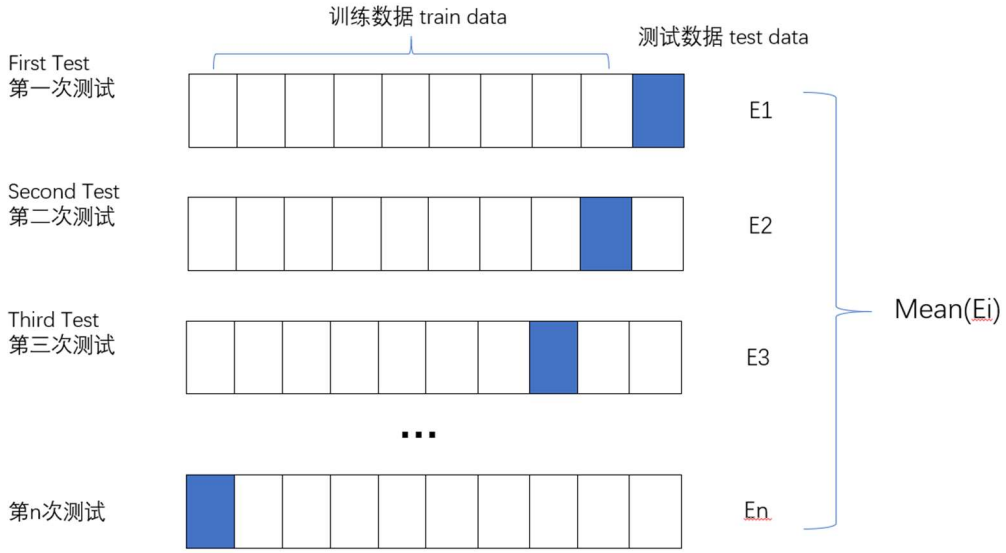


图 7.5 模型训练示意图

本文利用交叉验证的结果，采用网络搜索（Grid Search）的方式实现对模型参数的自动化调参。本研究调用 20 核 CPU 进行测试，每组测试均超过 3 小时该，过程耗时较长。最终通过交叉验证，我们得到具有更好预测效果的模型，图 X 表示是不同参数下模型预测指标的对比图 7.6。

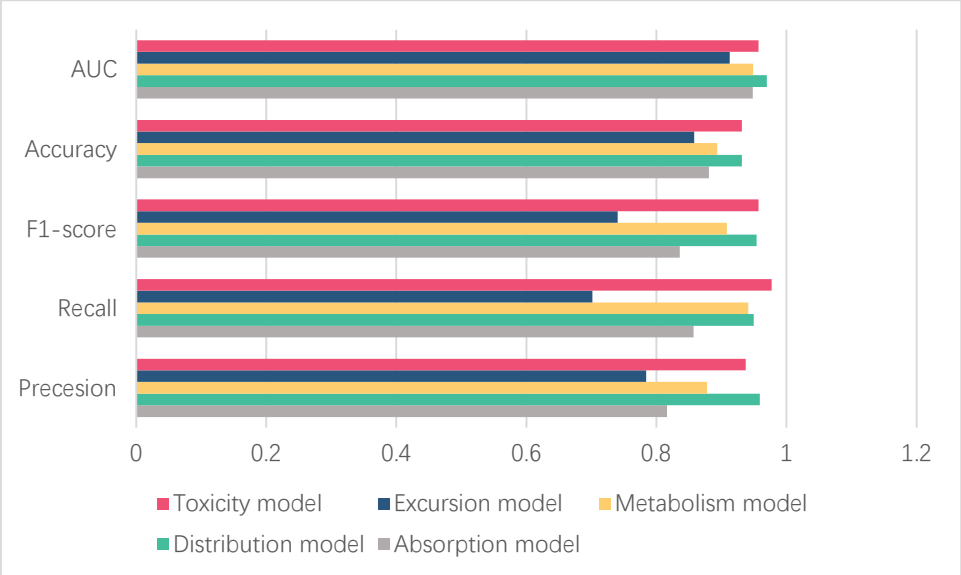


图 7.6 不同参数下模型预测指标的对比

7.3.3 最优阈值确定

二元-Logistic 回归计算得到样本的正概率，需要对概率设置阈值分类区间以区分真加类（1 和 0），如果概率高于阈值，则将样本分类为真。因此，不同样本的分类结果会随着阈值的改变而变化，本文增加了对最优阈值筛选的过程。

本文考虑几何平均数（G-Mean）作为确定最优阈值的评判指标。几何平均数是召回率和特异性这两个指标的几何平均数，写作：

$$G-Mean=\sqrt{Recall\times Specificity}=\sqrt{TPR\times\frac{TN}{FP+TN}}$$

其中，

$$\frac{TN}{FP+TN}=(1-\frac{FP}{FP+TN})=\sqrt{TPR\times(1-FPR)}$$

上式中的变量，即为在 sklearn 库中计算得到的变量，二者变量名相同，参见附件 C。这项措施试图最大限度地提高每个类的准确性，同时保持这些精度平衡。因此，通常采用它作为非平衡分类的无偏评价指标之一[6]。

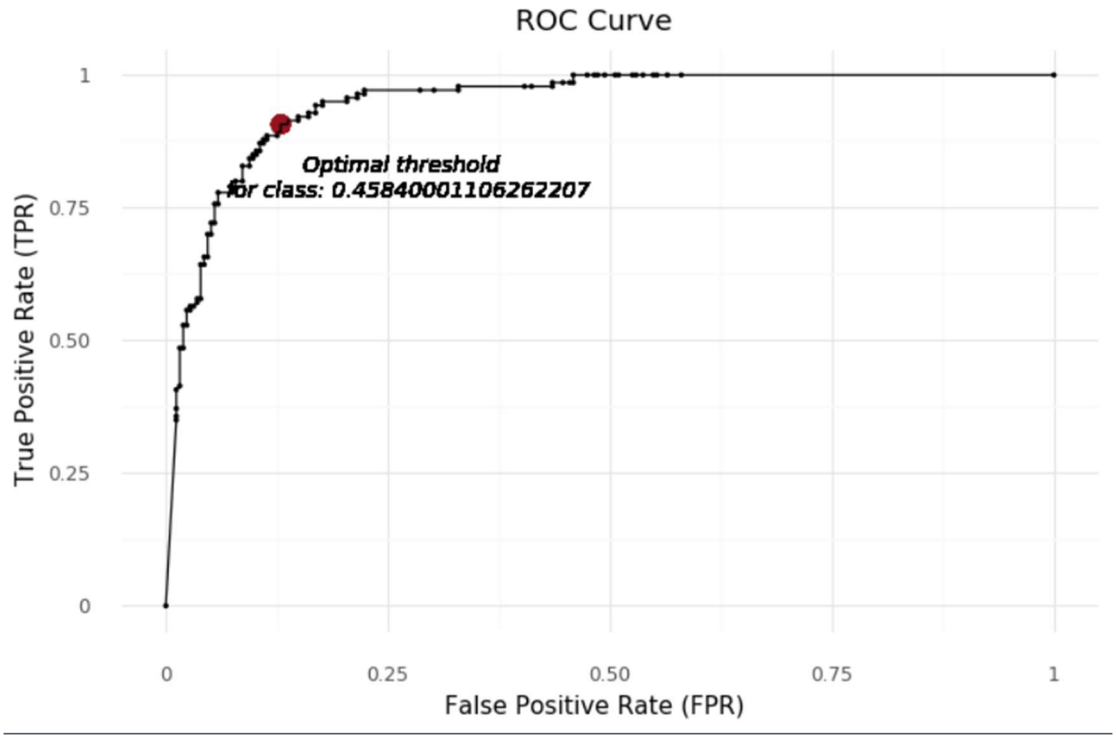


图 7.7 最优阈值确定示意图（以 Absorption model 为例）

以 G-Mean 作为无偏评价指标，以阈值移动为重点，生成二分类的最优阈值为 0.458。因此在理论上，当观测值的概率低于 0.458 时，将被归为次要类别，反之亦然。

优化阶段总结果由图 7.8 表示。可以看出的是，不同于经验方法采用 0.5 作为阈值标准，依据 G-Mean 作为评判标准其结果更加客观。

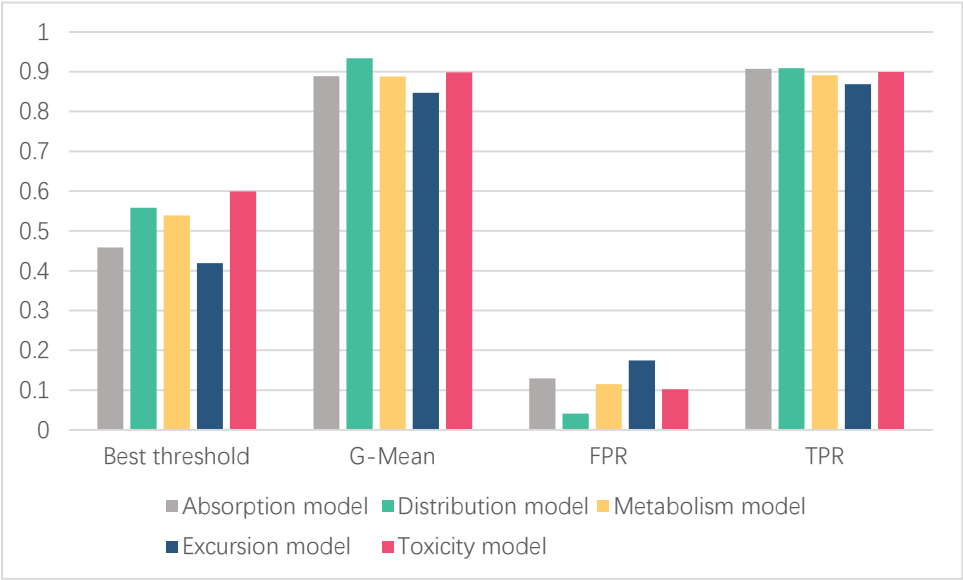


图 7.8 最优化结果

7.3.4 影响变量

分别计算得到对特征影响较大的五种特征重要性指标:

- ‘weight’：使用特征在所有树上拆分数据的次数。
- ‘gain’：使用该特征的所有分割的平均增益。
- ‘cover’：使用该特征的所有分割的平均覆盖率。
- ‘total_gain’：使用该特征的所有分割的总增益。
- ‘total_cover’：使用该特征的所有分割的总覆盖率。

下图反映的是不同特征变量下，ADMET 的影响程度大小。其中，仅有 ECCEN 一个变量同时对 Absorption 和 Metabolism 两个指标皆有较大影响。

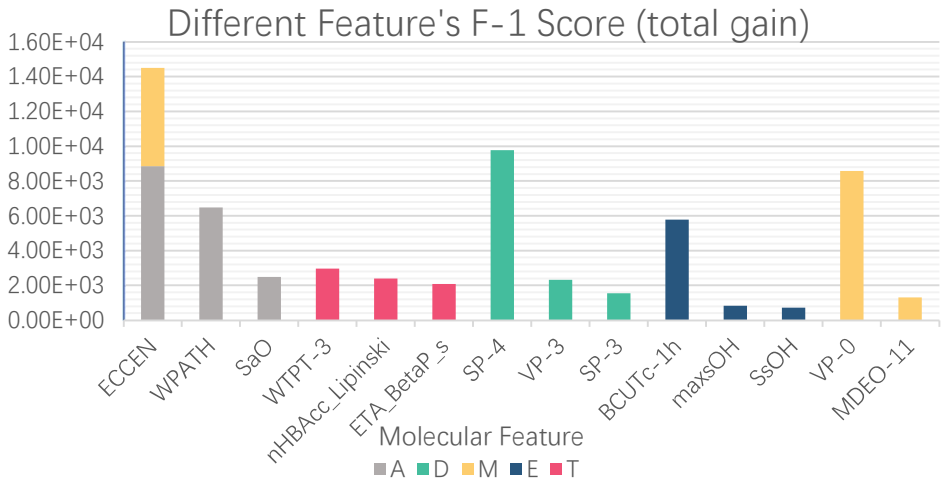


图 7.9 不同特征变量对 ADMET 结果的影响程度

7.4 结果展示

图 5 表示的依次吸收，分布，代谢，排泄，毒性（ADMET）五大指标的预测结果。为了更好的展示结果，本文将图表的结果用分别用多种颜色表示。在关于 ADMET 的指标文件中，数据仅为 1 和 0，因此本图按照医学惯例对这些数据进行阴性阳性分类。图中出现的点标记，均表示数值结果为 1，医学表现为阳性；同色点之间的留白，表示结果为阴性，数值结果为 0。

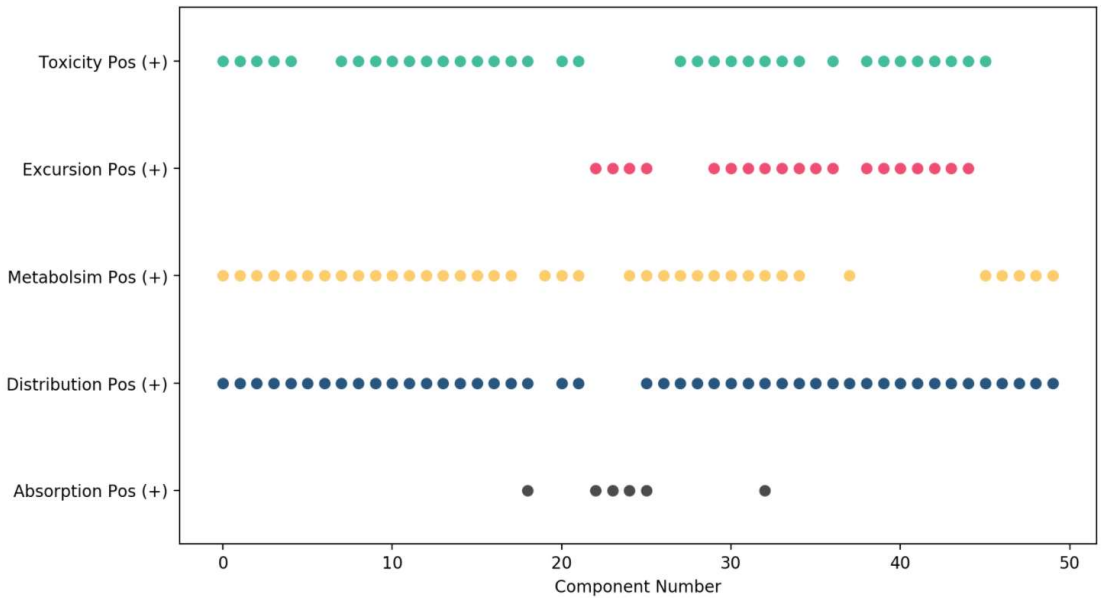


图 7.10 问题三预测结果的可视化

最后我们通过分类预测模型，对文件“ADMET.xlsx”的 test 表中的 50 个化合物进行相应的预测，并将结果填入“ADMET.xlsx”的 test 表中对应的 Caco-2、CYP3A4、hERG、HOB、MN 列。完成问题三要求。由于篇幅原因，预测见附录 2。

八、问题四（ERα 拮抗剂性质综合预测模型）

8.1 问题四分析

问题四要求，从分子描述符中寻找到合适的区间范围，要求该区间范围内分子描述符所影响的样本结果既满足生物活性的要求，又满足 ADMET 性质要求。

经过对前三个问题的研究和实现，目前的研究已经对影响生物活性和 ADMET 五大指标影响较大的分子描述符具有一定认识。在本节中所涉及的研究和讨论，均是基于前文所述的结果之上，提出的模型架构。

本文认为，在具有实际应用背景下的数据建模，其依据不应简单从这些数据的数值身上挖掘特征，还应考虑的是这些数据所代表的实际物理性质。就本文所讨论问题而言，探究化合物的药理特性能否符合题目所设要求是首要目标。因此，本文考虑将问题转化为多分类模型的构建问题，采用聚类思想对不同分子描述符内在特性进行挖掘探究。首先将题目所给 1984 条原始数据作为基本研究对象，对其 729 个分子描述变量进行数据降维，其依据是第一题中所得到 20 组对生物活性影响较大的特征变量和第三题中所得到 5 个指标影响较大的前 5 组特征变量（共计 40 组）。随后对于题目所提出的两点约束要求，本文考虑建立一个合理化的特征指标来对化合物能否满足以上两点进行评估。获得药物性分类指标后，对总体样本进行分类操作，获得多分类问题的目标数据集。构建好模型后，测试模型性能指标，模型用以预测任意变量下的化合物组分能否符合药物性指标，并且得到对模型性能影响最大的特征变量。最后利用聚类方法，对目标特征变量进行可视化聚类分析，完成特征变量范围的选取。

综上所述，本题的思考流程如下图 8.1 所示。

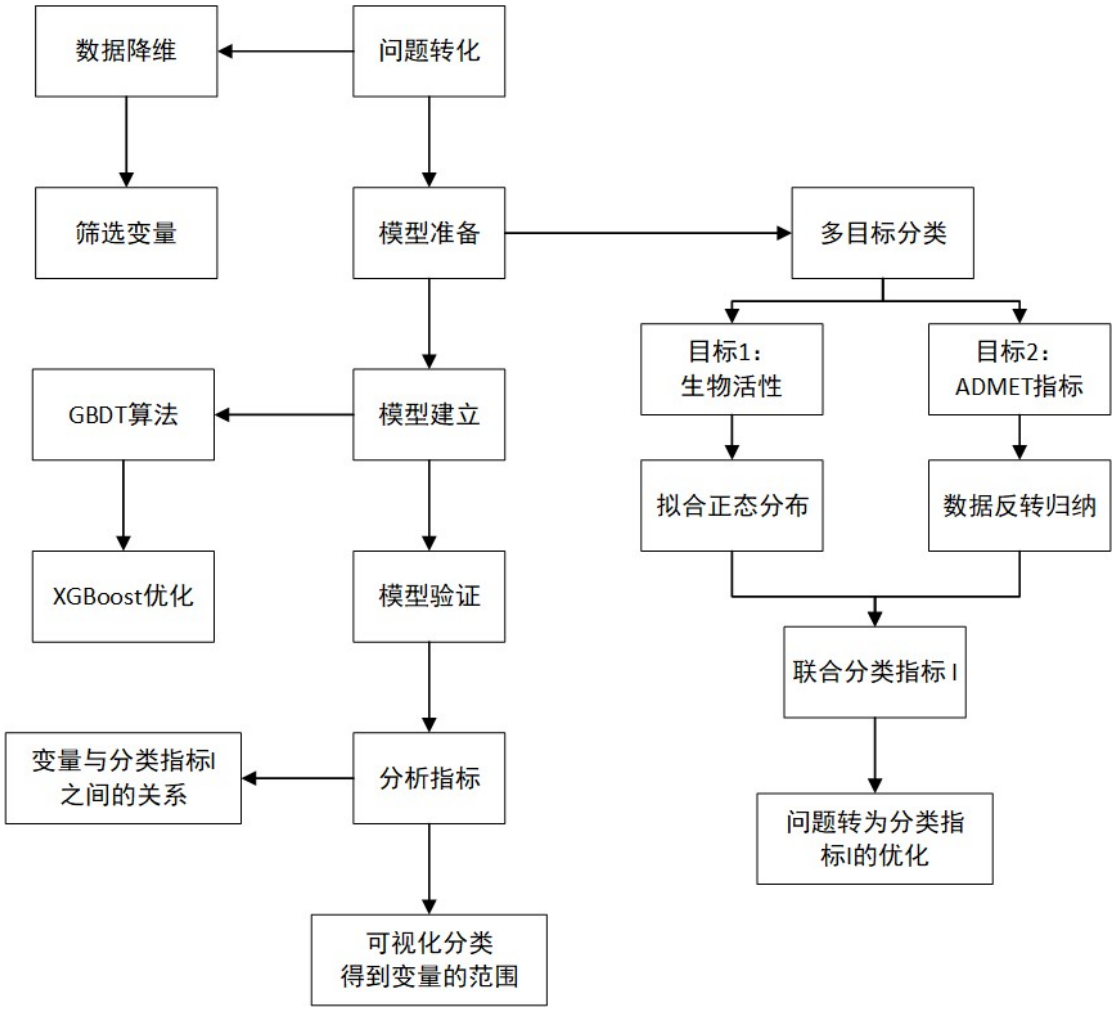


图 8.1 问题四模型求解流程图

8.2 问题四求解

8.2.1 数据降维

为了简化模型，减小计算时间，利用已有的结果来对总体数据进行数据降维是有必要的。本文对特征变量初步采用前文结果，从第一题所得到 20 组对生物活性影响较大的特征变量和第三题中所得到 5 个指标影响较大的前 5 组特征变量中选取，该过程即为初筛。

本文根据部分文献中总结的物理化学性质，进行解释和划分。表 8.1 描述的是对这些变量的物理解释。可以看出从各个特征变量的描述上，这些变量自身相关性不强。这进一步的验证了在第一题中，对几个特征变量的相关性研究是有效的。

表 8.1 特征变量的符号以及符号描述

符号	符号描述
LipoaffinityIndex	脂肪亲和力指数
C1SP2	与另一个碳结合的双链碳
maxssO	最大的原子型 E-状态： -O-

minHsOH	最小原子型 H E-状态:-OH
ndssC	原子型 E-状态的统计：=C<
SHBint10	路径长度为 10 的潜在氢键的 E-状态的强度统计
BCUTc-1h	最高部分电荷加权的 BCUTS
MDEC-23	所有二级和三级碳之间的分子边缘距离
VCH-5	效价链，顺序 5
minHBint10	路径长度为 10 的潜在氢键的最小 E-状态强度
MDEC-33	所有三级碳之间的分子边缘距离
WTPT-5	从氮原子开始的路径长度总和
MLFER_A	总体或总和溶质氢键酸度
SdssC	原子类型 E-状态之和：=C<
maxHBint5	路径长度的潜在氢键强度的最大 E-状态
ETA_Shape_Y	形状指数 Y
SPC-6	简单路径簇，顺序 6
minssCH2	最小原子型 E-状态：-CH2-
VCH-7	价链，顺序 7
BCUTp-1h	最高极化率加权的 BCUTS

8.2.2 分类指标

从问题描述上，提出既满足生物活性，又满足 ADMET 指标的要求。因此，在本题中的关键在于如何构建一个有效且定量的分类指标来对问题约束进行数学描述，将多目标优化问题转为单一目标问题。

优化目标一：生物活性

问题二中主要进行的是对生物活性的定量预测，从样本训练数据中，对 plC50 指标其进行正态分布拟合，结果如图所示。通过拟合，可以得到关于 plC50 值同概率密度的关系，即基于正态分布，可以对目标提出的较高生物活性指标 plC50 进行分类，其分类标准是根据正态分布的置信区间。当累计概率高于 15.5%，将其视为一般置信区间；当累计概率高于 84.25%，将其视为较好置信区间。基于此方法，对不连续的 plC50 进行合理化的预测，根据出现不同值时的累计概率密度，作为评价标准 A。

优化目标二：ADMET 指标

在关于 ADMET 的指标文件中，数据仅为 1 和 0，因此本文按照医学惯例对这些数据进行阴性阳性分类，其特性表现为是否对提升药物性筛选有利。值得注意的是，在 ADMET 五个指标中，1 并非全为有利指标，本文在处理上采用对

结果转置，最终获得同权的五个指标，通过计算五个指标之和，可以获得相应的分类区间。即，若一个样本的指标之和为 5，则判定为最优；若一个样本的指标之和为 4，则判定为次优；若一个样本的指标之和为 3，则判定为一般；若一个样本的指标之和为 2，则判定为不达标。其数学描述如下，记为评价标准 B

联合构建分类指标 I

为了更好的获得联合指标 I 来进行预测评价，考虑对这一指标进行区分，初始假设这些目标同等重要，用权值 w = 0.5 来对 A，B 指标平均化，最后增加对 ADMET 指标的约束，即对超过 3 的指标人为增加系数 1。

$$I = A * w + B * w + 1$$

从表中可看出，本节中提出的联合分类指标 I 的基本分类依据。为了更好的让数据结果直观展示，对表格绘制出性能分析图，不同颜色表示本文的三类基本分类。

表 8.2 联合分类指标分类

指标 A	指标 B	约束值	联合分类指标 I
0.5	0.8425	1	1.92125
0.5	0.156	1	1.617
0.4	0.8425	1	2.031875
0.4	0.156	1	1.517
0.3	0.8425	1	1.931875
0.3	0.156	1	1.417
0.2	0.8425	0	0.831875
0.2	0.156	0	0.317
0.1	0.8425	0	0.731875
0.1	0.156	0	0.217
0	0.8425	0	0.631875

从药理学角度阐明其含义为，绿色为最佳药物候选，具有较好的生物活性和基本实现 ADMET 指标；蓝色为一般药物候选，具有一般的生物活性和基本实现 ADMET 指标；红色分类为不达标药物。

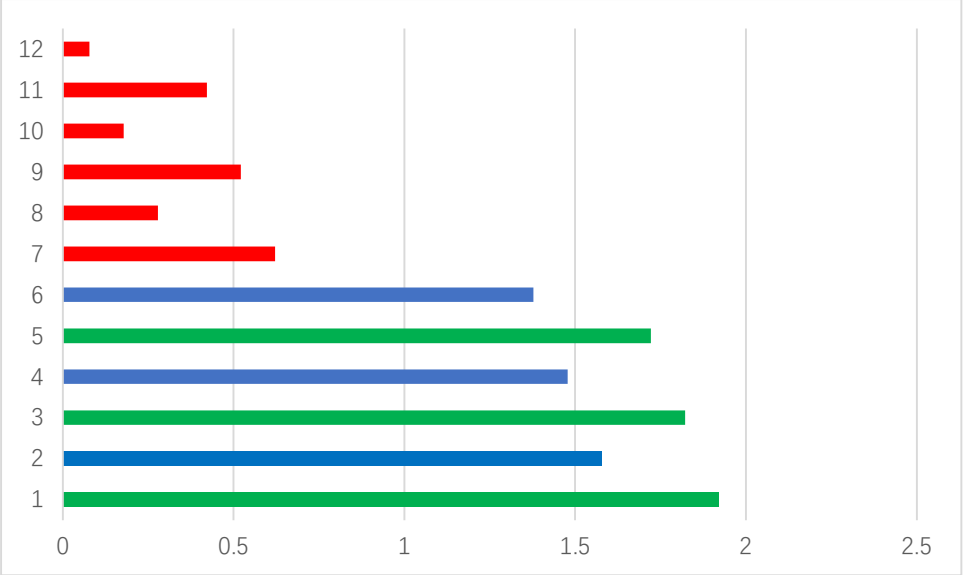


图 8.2 分类指标可视化

综上所述，本文采用构建了特征指标 I，并根据该指标下属不同权重的两种指标进行分类划分。最终采用的分类结果为，当计算的某一样本分类指标高于 1.721 时，视为最优组，其满足具有较好生物活性和实现 ADMET 指标的特征；当特征指标介于 1.721 和 1.378 时，其满足具有一般生物活性和实现 ADMET 指标的特征，视为一般组，最后，剩下的分类归为坏组，不参与讨论。

8.2.3 多分类模型

本文多分类预测模型同问题 3，均是通过 python 实现基于 xgboost 分类器的基本功能，相关代码见附件。其基本原理与实现，同上一节中对于 xgboost 的介绍。

在本节中，构建多分类预测模型的主要目的是探究影响较大的特征变量。通过累计评价指标 F1-Score，得到对该分类预测模型影响影响最显著的。

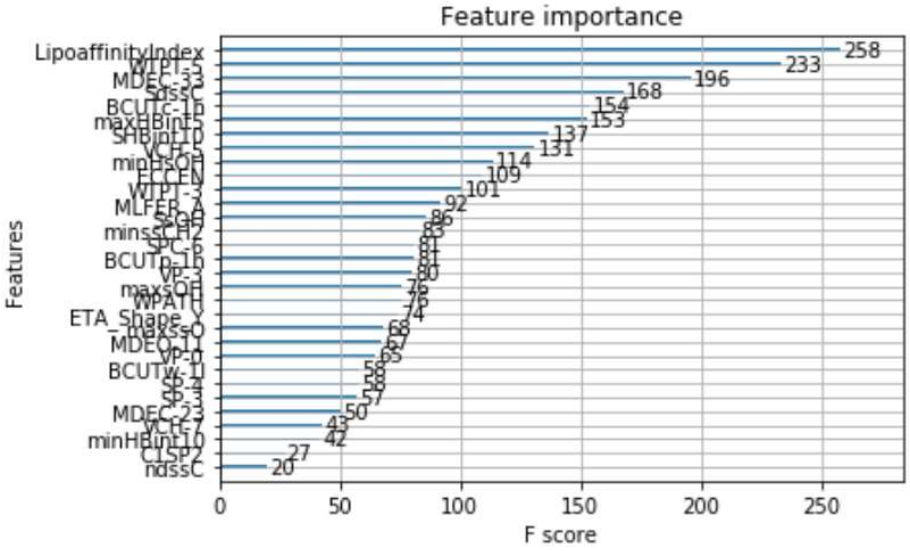


图 8.3 分类指标可视化

8.3 数据可视化

本节中，通过对数据进行可视化展示，凸显出其内在的数据特征。根据本节中提出的分类指标 I，对其进行聚合分析。图 8.4 展现的是主要四种变量对于样本的取值区间影响，不同颜色表示的是不同特征变量，横坐标展示的是样本编号索引。在结果中隐去了坏值组，仅展示最优组的部分。

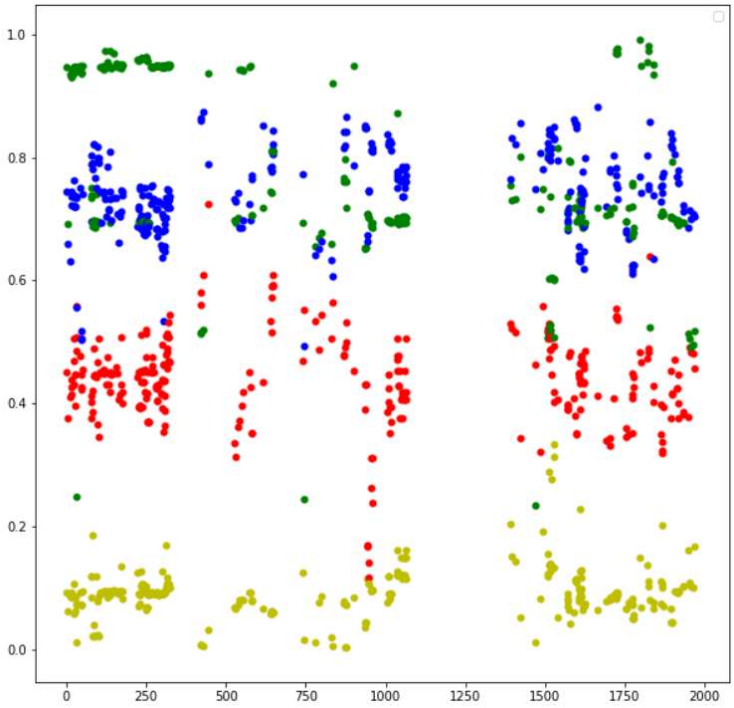


图 8.4 分类指标可视化散点图

通过对变量的筛选和分类，最终得到除去不符合要求的样本在主要影响指标I的统计学特征。基于这些统计学特征，我们对其再次进行正态分布的拟合，以概率分布为依据，绘制了相应的箱形图。其中，图 8.3 和 8.4 分别为分类指标 I 下较好和一般两类的箱型图，以及分类指标 I 下较好的箱型图。从这两张图中对比发现，主要变量的差异性并不明显。最终数据整理汇总至附件。

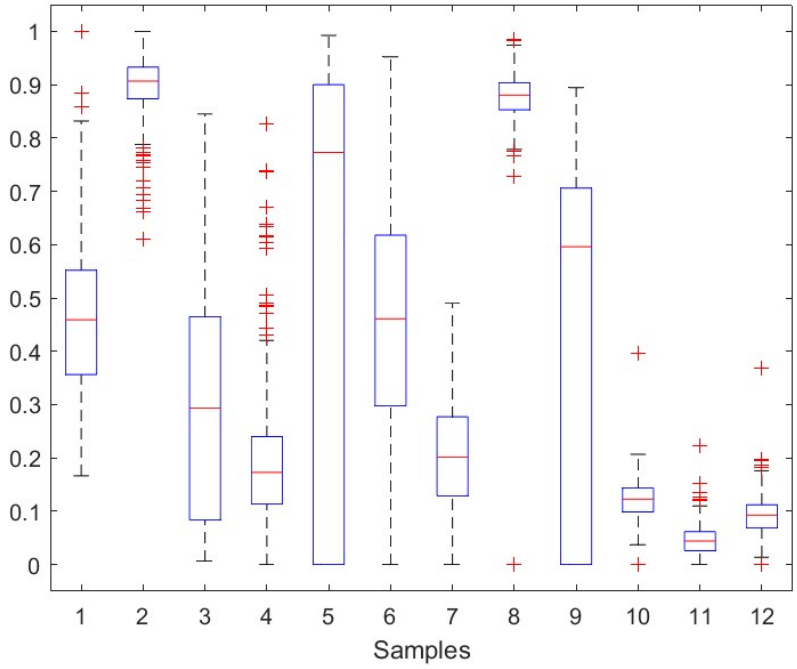


图 8.5 主要特征同分类指标的箱形图（包含最优和一般分类）

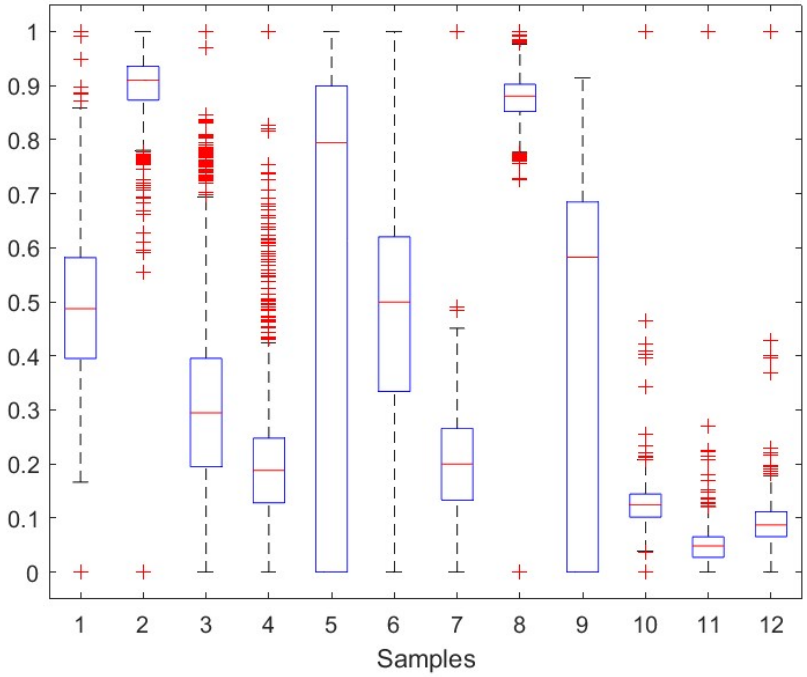
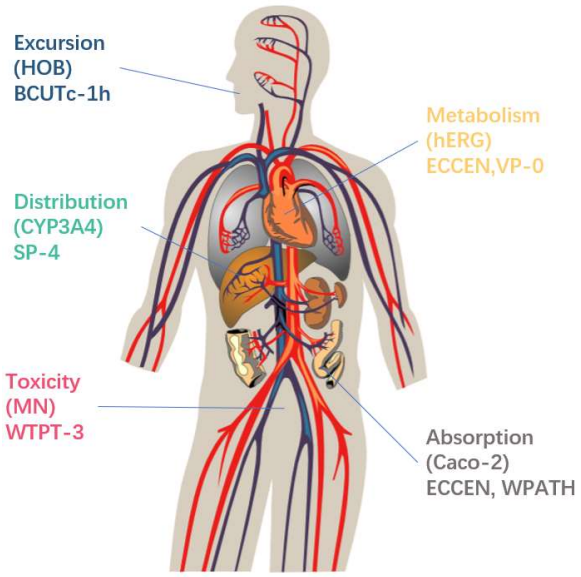


图 8.6 主要特征同分类指标的箱形图（仅包含最优分类）

8.4 小结

迫于时间关系和篇幅限制，只能对分类指标进行简要设计，在对本问题的概率预测提升上，还具备更多的优化空间。对于本节所讨论的数据和代码可参

加附录，最后是对本节结论的可视化展示，篇幅受限只展示了部分特征变量。



Estrogen receptors alpha
LipoaffinityIndex,
MDEC-33,
WTPT-5,
SdssC

Feature 特征变量	Ranges 取值范围	Mean 平均值
LipoaffinityIndex	(10.8, 16.2)	8.4
MDEC-33	(-0.2,-0.12)	-0.19
WTPT-5	(0.79, 5.2)	2.8
SdssC	(2.2, 4.8)	3.8
ECCEN	(380, 996)	626

图 8.7 结果可视化（问题四）

九、模型的评价与推广

9.1 模型的评价

（1）利用多种数学原理和计算软件，完成了数据处理

本文充分利用线性、非线性相关分析、随机森林算法、神经网络、GBDT 算法原理、遗传算法和 XGBoost 优化等数据挖掘技术研究了如何通过大量分子描述符分别对化合物活性和 ADMET 性质预测的问题。还建立评判判据，寻找到分子描述符在合适取值范围时，能够使化合物对抑制 ERα 具有更好的生物活性，同时具有更好的 ADMET 性质。

（2）本文在研究问题时考虑较为全面

针对问题一进行了高相关性滤波，针对问题二进行了五种算法的对比法分析。针对问题四进行了人体健康综合性考量。分析与对比不但保证了文章的严谨性，特别是针对实际药用场景应用，更需要对所建立的模型以及结果进行多种方法对比优化，保证预测的准确性，力求给药物研发以指导方向。

9.2 模型的推广

在后续研究中，为保证同时兼顾数据的深度与广度，需要提高样本数量，对数据集进行多次训练，不断优化模型，提升模型预测精度。对于本文筛选出的主要变量，并未分析其在模型中的具体表现的生物化学联系，因此相关药物研发人员可考虑使用本文的预测模型进行进一步的内在联系模式的探究。

本文主要是针对乳腺癌治疗靶标 ERα 研究分子描述符信息对化合物活性以及 ADMET 性质影响下的预测与优化，但是建立的模型也可以预测与优化其他癌症的类似靶标，即本文的模型在医药研发领域具有相当的推广价值。

参考文献

[1] 黄育北.中国女性乳腺癌筛查指南[J].中国肿瘤临床,2019(09):429-431.
[2]Luan F, Liu H, Wen Y, et al. QSPR Study for Estimation of Density of Some Aromatic Explosives by Mul
[3] Slater O, Kontoyianni M. The compromise of virtual screening and its impact on drug discovery[J]. Expert opinion on drug discovery, 2019, 14(7): 619-637.
[4]魏星,胡德华,易敏寒,常雪莲,朱文婕,曲少玲,邓端英.乳腺癌基因药物网络模型的构建与分析[J].南方医科大学学报,2016,36(02):170-179.
[5]中国抗癌协会乳腺癌专业委员会.中国抗癌协会乳腺癌诊治指南与规范(2017年版)[J].中国癌症杂志,2017,27(09):695-759.
[6]周志华, 机器学习[M], 清华大学出版社, 2016.01
[7]Torres Barran A, Alonso A, Dorronsoro J R. Regression tree ensembles for wind energy and solar radiation prediction[J]. Neurocomputing. 2018, 326: 151 160.
[8] Study of Peak Load Demand Estimation Methodology by Pearson Correlation Analysis with Macro-economic Indices and Power Generation Considering Power Supply Interruption. Journal of Electrical Engineering & Technology, 2017,12(4，
[9]XGBoost: A Scalable Tree Boosting System
[10]胡军,张超,陈平雁.非参数双变量相关分析方法 Spearman 和 Kendall 的模拟比较[J].中国卫生统计,2008,25(06):590-591.
[11] P. R D L, Fatichah C, Purwitasari D. Deteksi Gempa Berdasarkan Data Twitter Menggunakan Decision Tree, Random Forest, dan SVM. Jurnal Teknik ITS, 2017,6(1).
[12]Agajanian S, Oluyemi O, Verkhivker G M, et al. Integration of Random Forest Classifiers and Deep Convolutional Neural Networks for Classification and Biomolecular Modeling of Cancer Driver Mutations. Frontiers in Molecular Biosciences, 2019,6.
[13] Chen T， Tong H， Benesty M . xgboost: Extreme Gradient Boosting[J]. 2016.

附录

附录 1：“ERα_activity.xlsx”的 test 表中的 IC50_nM 列及对应的 pIC50 列展示。

说明：由于化合物结构的 SMILES 式过长，为了简便美观，把 test 表的第 X 个化合物的 SMILES 式用 Sample X 表示，如第一个化合物 SMILES 式 COc1cc(OC)cc(\C=C\c2ccc(OS(=O)(=O)[C@@H]3C[C@@H]4O[C@H]3C(=C4c5ccc(O)cc5)c6ccc(O)cc6)cc2)c1 记作 Sample 1。

SMILES	IC50_nM	pIC50		SMILES	IC50_nM	pIC50
Sample1	119.60764	6.92224		Sample26	919.8013	6.03631
Sample2	44.80954	7.34863		Sample27	120.04459	6.92066
Sample3	19.086023	7.71928		Sample28	3147.6179	5.50202
Sample4	17.45583	7.75806		Sample29	4029.7674	5.39472
Sample5	7.6577008	8.1159		Sample30	2233.7234	5.65097
Sample6	4.7341518	8.32476		Sample31	14709.908	4.83239
Sample7	10.501192	7.97876		Sample32	9474.9325	5.02342
Sample8	11.898756	7.9245		Sample33	14547.075	4.83722
Sample9	39.665	7.402		Sample34	5382.0018	5.26906
Sample10	8.8280884	8.054		Sample35	36226.865	4.44097
Sample11	7.1423269	8.14616		Sample36	0.294577	9.5308
Sample12	31.757672	7.49815		Sample37	0.378458	9.42198
Sample13	131.66637	6.88053		Sample38	9393.3611	5.02718
Sample14	14.201662	7.84766		Sample39	99020.373	4.00428
Sample15	28.498626	7.54518		Sample40	98001.859	4.00877
Sample16	10.042956	7.99814		Sample41	98072.841	4.00845
Sample17	21.236141	7.67292		Sample42	98045.156	4.00857
Sample18	87.184891	7.05956		Sample43	98247.626	4.00768
Sample19	9.3667954	8.02841		Sample44	98088.189	4.00838

Sample20	45.017448	7.34662		Sample45	98072.841	4.00845
Sample21	184.53374	6.73392		Sample46	14.452285	7.84006
Sample22	136.36977	6.86528		Sample47	17.5803	7.75497
Sample23	18.041141	7.74374		Sample48	15.126189	7.82027
Sample24	13.848456	7.8586		Sample49	33.468856	7.47536
Sample25	3825.7318	5.41729		Sample50	6.0806876	8.21605

附录 2：“ADMET.xlsx”的 test 表中对应的 Caco-2、CYP3A4、hERG、HOB、MN50 列展示。

说明：化合物结构的 SMILES 式用 Sample X 表示，具体见附录 1 说明

SMILES	Caco-2	CYP3A4	hERG	HOB	MN
Sample1	0	1	1	0	1
Sample2	0	1	1	0	1
Sample3	0	1	1	0	1
Sample4	0	1	1	0	1
Sample5	0	1	1	0	1
Sample6	0	1	1	0	0
Sample7	0	1	1	0	0
Sample8	0	1	1	0	1
Sample9	0	1	1	0	1
Sample10	0	1	1	0	1
Sample11	0	1	1	0	1
Sample12	0	1	1	0	1
Sample13	0	1	1	0	1
Sample14	0	1	1	0	1
Sample15	0	1	1	0	1
Sample16	0	1	1	0	1
Sample17	0	1	1	0	1
Sample18	0	1	1	0	1
Sample19	1	1	0	0	1
Sample20	0	0	1	0	0
Sample21	0	1	1	0	1
Sample22	0	1	1	0	1

Sample23	1	0	0	1	0
Sample24	1	0	0	1	0
Sample25	1	0	1	1	0
Sample26	1	1	1	1	0
Sample27	0	1	1	0	0
Sample28	0	1	1	0	1
Sample29	0	1	1	0	1
Sample30	0	1	1	1	1
Sample31	0	1	1	1	1
Sample32	0	1	1	1	1
Sample33	1	1	1	1	1
Sample34	0	1	1	1	1
Sample35	0	1	1	1	1
Sample36	0	1	0	1	0
Sample37	0	1	0	1	1
Sample38	0	1	1	0	0
Sample39	0	1	0	1	1
Sample40	0	1	0	1	1
Sample41	0	1	0	1	1
Sample42	0	1	0	1	1
Sample43	0	1	0	1	1
Sample44	0	1	0	1	1
Sample45	0	1	0	1	1
Sample46	0	1	1	0	1
Sample47	0	1	1	0	0

Sample48	0	1	1	0	0
Sample49	0	1	1	0	0
Sample50	0	1	1	0	0

附录 3：本文建立模型所编写的主要程序展示

程序编号	1	程序说明	问题 1 代码	编写软件	Matlab

```
D_Q1.m
%% Data Load
MD_train=xlsread('Molecular_Descriptor.xlsx',1,'B1:ABB1975')
ER_target=xlsread('ERα_activity.xlsx',1,'B1:B2:C1975')
ADMET_training=xlsread('ADMET','training')
%MolecularDescriptor = improtfile("Molecular_Descriptor.xlsx", "training", 'B2:ABB1975')
% Molecular = importfile("Molecular_Descriptor.xlsx", "training", 'A2:A1975')
load('MolecularDescriptor.mat')
MD_test=xlsread('Molecular_Descriptor.xlsx',2,'B1:ABB51')
%% 数据预处理

idx0=find(all(MD_train==0))
%idx_yc=find(ER_target(:,2)<4)
MD1=MD_train
ER1=ER_target
MD1(:,idx0)=[]
%ER1(idx_yc,:)=[]
MD_idx0=MolecularDescriptor
MD_idx0(:,idx0)=[]
%MD1(idx_yc,:)=[]
Mo_af_del=Molecular
%Mo_af_del(idx_yc,:)=[]
%% 随机森林

y_exact=ER1(:,2)
x_train=MD1
Mdl1= TreeBagger(200,x_train,y_exact,'Method','regression','Surrogate','on' ...
   ,'PredictorSelection','curvature','OOBPredictorImportance','on')
imp = Mdl1.OOBPermutedPredictorDeltaError
%%
[impd idx1]=sort(abs(imp),'descend');
%%
figure
h=barh(impd(:,1:30),'LineWidth',1)
title('Standard CART');
xlabel('Predictor importance estimates');
ylabel('Predictors');
ticks_y=MD_idx0(idx1(1,1:30))
yticks([1:30])
yticklabels(ticks_y)
set(gca,'YDir','reverse')
%% 变量相关性分析

[R,P,RL,RU] = corrcoef(MD1(:,idx1(1,1:30)))
heatmap(MD_idx0(1,idx1(1,1:30)),MD_idx0(1,idx1(1,1:30)),R.^2)
colormap(flipud(pink))
```

```
[raw col]=find(R.^2>0.6)
del=[raw col]
% for i=1:size(del,1)
%     if del(i,1)==del(i,2)
%         del(i,:)=[]
%     end
% end
title('Correlation Analysis')
%%
del_num=[4 17 23 21 8 28 29]
MD2=MD1(:,idx1(1,1:30))
MD_idx2=MD_idx0(1,idx1(1,1:30))
MD2(:,del_num)=[]
MD_idx2(:,del_num)=[]
MD2=MD2(:,1:20)
MD_idx2=MD_idx2(:,1:20)
R1 = corrcoef(MD2)
heatmap(MD_idx2,MD_idx2,R1.^2)
colormap(flipud(pink))
title('Correlation Analysis')
%%
%数据保存
SMILES=Mo_af_del
MD_idx22 = cellstr(MD_idx2)
% T1=table([MD_idx2 MD2],'Rowname',)
% T2=table(SMILES,ER1(:,1),ER1(:,2))
% T2.Properties.VariableNames = {'SMILES','ic50','pic50'}
```

程序编号	2	程序说明	问题 2 代码	编写软件	Matlab

```
GABPnet.m
% 此代码调用的部分函数来自 https://github.com/UoS-CODeM/GA-Toolbox
% clc
% clear all
% close all
%% 加载神经网络的训练样本 测试样本每列一个样本 输入 P 输出 T
%样本数据就是前面问题描述中列出的数据
%load data
% 初始隐层神经元个数
hiddennum=25;
% 输入向量的最大值和最小值
threshold=[0 1;0 1;0 1;0 1;0 1;0 1;0 1;0 1;0 1;0 1;0 1;0 1;0 1;0 1];
inputnum=size(P,1);       % 输入层神经元个数
outputnum=size(T,1);      % 输出层神经元个数
w1num=inputnum*hiddennum; % 输入层到隐层的权值个数
w2num=outputnum*hiddennum;% 隐层到输出层的权值个数
N=w1num+hiddennum+w2num+outputnum; %待优化的变量的个数

%% 定义遗传算法参数
NIND=40;        %个体数目
MAXGEN=50;     %最大遗传代数
```

```
PRECI=10;      %变量的二进制位数
GGAP=0.95;     %代沟
px=0.7;        %交叉概率
pm=0.01;       %变异概率
trace=zeros(N+1,MAXGEN);                    %寻优结果的初始值

FieldD=[repmat(PRECI,1,N);repmat([-0.5;0.5],1,N);repmat([1;0;1;1],1,N)];                    %区域描述器
Chrom=crtbp(NIND,PRECI*N);                  %初始种群
%% 优化
gen=0;                              %代计数器
X=bs2rv(Chrom,FieldD);              %计算初始种群的十进制转换
ObjV=Objfun(X,P,T,hiddennum,p_test,T_test);       %计算目标函数值
while gen<MAXGEN
  fprintf('%d\n',gen)
  FitnV=ranking(ObjV);                          %分配适应度值
  SelCh=select('sus',Chrom,FitnV,GGAP);           %选择
  SelCh=recombin('xovsp',SelCh,px);               %重组
  SelCh=mut(SelCh,pm);                            %变异
  X=bs2rv(SelCh,FieldD);            %子代个体的十进制转换
  ObjVSel=Objfun(X,P,T,hiddennum,p_test,t_test);          %计算子代的目标函数值
  [Chrom,ObjV]=reins(Chrom,SelCh,1,1,ObjV,ObjVSel); %重插入子代到父代，得到新种群
  X=bs2rv(Chrom,FieldD);
  gen=gen+1;                                    %代计数器增加
  %获取每代的最优解及其序号，Y 为最优解,I 为个体的序号
  [Y,I]=min(ObjV);
  trace(1:N,gen)=X(I,:);                    %记下每代的最优值
  trace(end,gen)=Y;                          %记下每代的最优值
end
%% 画进化图
figure(1);
plot(1:MAXGEN,trace(end,:));
grid on
xlabel('遗传代数')
ylabel('误差的变化')
title('进化过程')
bestX=trace(1:end-1,end);
bestErr=trace(end,end);
fprintf(['最优初始权值和阈值:\nX=',num2str(bestX'),'\n 最小误差 err=',num2str(bestErr),'\n'])


%% 比较优化前后的训练&测试
callbackfun

callbackfun.m
Train_obj=MD2
Traget_obj=ER1(:,2)
% resultfrom=iMdtest(:,RFindex(:,1:20))
TT_obj=[Train_obj Traget_obj]
%% 不使用遗传算法
% 使用随机权值和阈值
inputnum=size(P,1);      % 输入层神经元个数
outputnum=size(T,1);     % 输出层神经元个数
%% 归一化
```

```
temp=randperm(size(TT_obj,1))%打乱样本排序
P_train=TT_obj(temp(1:1700),1:size(Train_obj,2))'
T_train=TT_obj(temp(1:1700),size(Train_obj,2)+1)'
P_test=TT_obj(temp(1701:end),1:1:size(Train_obj,2))'
T_test=TT_obj(temp(1701:end),size(Train_obj,2)+1)'
N=size(P_test,2);
[p_train,ps_input]=mapminmax(P_train,0,1)
p_test=mapminmax('apply',P_test,ps_input)
[t_train,ps_output]=mapminmax(T_train,0,1)
%t_test=mapminmax('apply',T_test,ps_output)
%% 新建 BP 网络
net=newff(minmax(P),[hiddennum,outputnum],{'tansig','logsig'},'trainlm');
%% 设置网络参数：训练次数为 1000，训练目标为 0.01，学习速率为 0.1
net.trainParam.epochs=5000;
net.trainParam.goal=0.001;
LP.lr=0.01;
%% 训练网络以
netBP=train(net,P,T);
%% 测试网络
disp(['1、使用随机权值和阈值 '])
disp('测试样本预测结果：')
t_sim1=sim(netBP,p_test)
T_sim1=mapminmax("reverse",t_sim1,ps_output)
err1=sum(abs(T_sim1-T_test)./T_test)/size(T_sim1,2);   %测试样本的仿真误差
err11=sum(abs(mapminmax("reverse",sim(netBP,p_train),ps_output)-T_train)./T_train)/size(T_train,2);   %
训练样本的仿真误差
disp(['测试样本的仿真误差:',num2str(err1)])
disp(['训练样本的仿真误差:',num2str(err11)])


%% 使用遗传算法
%% 使用优化后的权值和阈值
inputnum=size(P,1);     % 输入层神经元个数
outputnum=size(T,1);    % 输出层神经元个数
%% 新建 BP 网络
net=newff(minmax(P),[hiddennum,outputnum],{'tansig','logsig'},'trainlm');
%% 设置网络参数：训练次数为 1000，训练目标为 0.01，学习速率为 0.1
net.trainParam.epochs=5000;
net.trainParam.goal=0.01;
LP.lr=0.01;
%% BP 神经网络初始权值和阈值
w1num=inputnum*hiddennum; % 输入层到隐层的权值个数
w2num=outputnum*hiddennum;% 隐层到输出层的权值个数
w1=bestX(1:w1num);  %初始输入层到隐层的权值
B1=bestX(w1num+1:w1num+hiddennum);  %初始隐层阈值
w2=bestX(w1num+hiddennum+1:w1num+hiddennum+w2num); %初始隐层到输出层的阈值
B2=bestX(w1num+hiddennum+w2num+1:w1num+hiddennum+w2num+outputnum); %输出层阈值
net.iw{1,1}=reshape(w1,hiddennum,inputnum);
net.lw{2,1}=reshape(w2,outputnum,hiddennum);
net.b{1}=reshape(B1,hiddennum,1);
net.b{2}=reshape(B2,outputnum,1);
%% 训练网络以
netGABP=train(net,P,T);
%% 测试网络
```

```
disp(['2、使用优化后的权值和阈值'])
disp('测试样本预测结果：')
t_sim2=sim(netGABP,p_test)
T_sim2=mapminmax("reverse",t_sim2,ps_output)
err2=sum(abs(T_sim2-T_test)./T_test)/size(T_test,2);
err21=sum(abs(mapminmax("reverse",sim(netGABP,p_train),ps_output)-T_train)./T_train)/size(T_train,2);
disp(['测试样本的仿真误差:',num2str(err2)])
disp(['训练样本的仿真误差:',num2str(err21)])
%% 预测值
mdtest_minmax=mapminmax('apply',MDtest1.',ps_input)
Y_result=mapminmax("reverse",sim(netGABP,mdtest_minmax),ps_output).'
%%
%R^2
RR1=1-sum((T_sim1-T_test).^2)/sum((T_test-sum(T_test)/size(T_test,2)).^2)
RR2=1-sum((T_sim2-T_test).^2)/sum((T_test-sum(T_test)/size(T_test,2)).^2)
%mse
mse1=sum((T_sim1-T_test).^2)/size(T_test,2)
mse2=sum((T_sim2-T_test).^2)/size(T_test,2)
%rmse
rmse1=sqrt(mse1)
rmse2=sqrt(mse2)
%mae
mae1=sum(abs(T_sim1-T_test))/size(T_test,2)
mae2=sum(abs(T_sim2-T_test))/size(T_test,2)
```

trainRegressionModel_GPR.m

```
%本程序基于 Regression Learner 工具箱
function [trainedModel, validationRMSE] = trainRegressionModel_GPR(trainingData, responseData)
% 将输入转换为表
inputTable = array2table(trainingData, 'VariableNames', ["LipoaffinityIndex","nHBAcc","minHsOH","maxssO","ndssC","MDEC-23",...
    "MDEC-33","VCH-5","ETA_BetaP_s","SHBint10","ATSc2","minHBa","maxHsOH","VPC-6","MLFER_S","minHBint10","MLFER_A","ETA_Shape_Y","SCH-5","TopoPSA"]);

predictorNames = ["LipoaffinityIndex","nHBAcc","minHsOH","maxssO","ndssC","MDEC-23",...
    "MDEC-33","VCH-5","ETA_BetaP_s","SHBint10","ATSc2","minHBa","maxHsOH","VPC-6","MLFER_S","minHBint10","MLFER_A","ETA_Shape_Y","SCH-5","TopoPSA"];
predictors = inputTable(:, predictorNames);
response = responseData(:);
isCategoricalPredictor = [false, false, false, false, false, false, false, false, false, false, false, false, false, false, false, false, false, false, false, false];

% 训练回归模型
% 以下代码指定所有模型选项并训练模型。
regressionGP = fitrgp(...
    predictors, ...
    response, ...
    'BasisFunction', 'constant', ...
    'KernelFunction', 'rationalquadratic', ...
    'Standardize', true);

% 使用预测函数创建结果结构体
predictorExtractionFcn = @(x) array2table(x, 'VariableNames', predictorNames);
gpPredictFcn = @(x) predict(regressionGP, x);
trainedModel.predictFcn = @(x) gpPredictFcn(predictorExtractionFcn(x));
```

```
% 向结果结构体中添加字段
trainedModel.RegressionGP = regressionGP;


% 提取预测变量和响应
% 以下代码将数据处理为合适的形状以训练模型。
%
% 将输入转换为表
inputTable = array2table(trainingData, 'VariableNames',
["LipoaffinityIndex","nHBAcc","minHsOH","maxssO","ndssC","MDEC-23",...
    "MDEC-33","VCH-5","ETA_BetaP_s","SHBint10","ATSc2","minHBa","maxHsOH","VPC-
6","MLFER_S","minHBint10","MLFER_A","ETA_Shape_Y","SCH-5","TopoPSA"];

predictorNames = ["LipoaffinityIndex","nHBAcc","minHsOH","maxssO","ndssC","MDEC-23","MDEC-
33","VCH-5","ETA_BetaP_s","SHBint10","ATSc2","minHBa","maxHsOH",...
    "VPC-6","MLFER_S","minHBint10","MLFER_A","ETA_Shape_Y","SCH-5","TopoPSA"];
predictors = inputTable(:, predictorNames);
response = responseData(:);
isCategoricalPredictor = [false, false, false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false];

% 执行交叉验证
partitionedModel = crossval(trainedModel.RegressionGP, 'KFold', 10);

% 计算验证预测
validationPredictions = kfoldPredict(partitionedModel);

% 计算验证 RMSE
validationRMSE = sqrt(kfoldLoss(partitionedModel, 'LossFun', 'mse'));
```

trainRegressionModel_SVM.m

```
%本程序基于 Regression Learner 工具箱
function [trainedModel, validationRMSE] = trainRegressionModel_SVM(trainingData, responseData)

% 将输入转换为表
inputTable = array2table(trainingData,
'VariableNames',["LipoaffinityIndex","nHBAcc","minHsOH","maxssO","ndssC","MDEC-23",...
    "MDEC-33","VCH-5","ETA_BetaP_s","SHBint10","ATSc2","minHBa","maxHsOH","VPC-
6","MLFER_S","minHBint10","MLFER_A","ETA_Shape_Y","SCH-5","TopoPSA"]);

predictorNames = ["LipoaffinityIndex","nHBAcc","minHsOH","maxssO","ndssC","MDEC-23",...
    "MDEC-33","VCH-5","ETA_BetaP_s","SHBint10","ATSc2","minHBa","maxHsOH","VPC-
6","MLFER_S","minHBint10","MLFER_A","ETA_Shape_Y","SCH-5","TopoPSA"];
predictors = inputTable(:, predictorNames);
response = responseData(:);
isCategoricalPredictor = [false, false, false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false];

% 训练回归模型
% 以下代码指定所有模型选项并训练模型。
responseScale = iqr(response);
if ~isfinite(responseScale) || responseScale == 0.0
    responseScale = 1.0;
end
boxConstraint = responseScale/1.349;
epsilon = responseScale/13.49;
```

```
regressionSVM = fitrsvm(...
    predictors, ...
    response, ...
    'KernelFunction', 'gaussian', ...
    'PolynomialOrder', [], ...
    'KernelScale', 4.5, ...
    'BoxConstraint', boxConstraint, ...
    'Epsilon', epsilon, ...
    'Standardize', true);

% 使用预测函数创建结果结构体
predictorExtractionFcn = @(x) array2table(x, 'VariableNames', predictorNames);
svmPredictFcn = @(x) predict(regressionSVM, x);
trainedModel.predictFcn = @(x) svmPredictFcn(predictorExtractionFcn(x));

% 向结果结构体中添加字段
trainedModel.RegressionSVM = regressionSVM;

% 提取预测变量和响应
% 以下代码将数据处理为合适的形状以训练模型。
%
% 将输入转换为表
inputTable = array2table(trainingData, 'VariableNames',
["LipoaffinityIndex","nHBAcc","minHsOH","maxssO","ndssC","MDEC-23",...
    "MDEC-33","VCH-5","ETA_BetaP_s","SHBint10","ATSc2","minHBa","maxHsOH","VPC-
6","MLFER_S","minHBint10","MLFER_A","ETA_Shape_Y","SCH-5","TopoPSA"]);

predictorNames = ["LipoaffinityIndex","nHBAcc","minHsOH","maxssO","ndssC","MDEC-23",...
    "MDEC-33","VCH-5","ETA_BetaP_s","SHBint10","ATSc2","minHBa","maxHsOH","VPC-
6","MLFER_S","minHBint10","MLFER_A","ETA_Shape_Y","SCH-5","TopoPSA"];
predictors = inputTable(:, predictorNames);
response = responseData(:);
isCategoricalPredictor = [false, false, false, false, false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false];

% 执行交叉验证
KFolds = 10;
cvp = cvpartition(size(response, 1), 'KFold', KFolds);
% 将预测初始化为适当的大小
validationPredictions = response;
for fold = 1:KFolds
    trainingPredictors = predictors(cvp.training(fold), :);
    trainingResponse = response(cvp.training(fold), :);
    foldIsCategoricalPredictor = isCategoricalPredictor;

    % 训练回归模型
    % 以下代码指定所有模型选项并训练模型。
    responseScale = iqr(trainingResponse);
    if ~isfinite(responseScale) || responseScale == 0.0
        responseScale = 1.0;
    end
    boxConstraint = responseScale/1.349;
    epsilon = responseScale/13.49;
    regressionSVM = fitrsvm(...
        trainingPredictors, ...
        trainingResponse, ...
```

```
        'KernelFunction', 'gaussian', ...
        'PolynomialOrder', [], ...
        'KernelScale', 4.5, ...
        'BoxConstraint', boxConstraint, ...
        'Epsilon', epsilon, ...
        'Standardize', true);

    % 使用预测函数创建结果结构体
    svmPredictFcn = @(x) predict(regressionSVM, x);
    validationPredictFcn = @(x) svmPredictFcn(x);

    % 向结果结构体中添加字段

    % 计算验证预测
    validationPredictors = predictors(cvp.test(fold), :);
    foldPredictions = validationPredictFcn(validationPredictors);

    % 按原始顺序存储预测
    validationPredictions(cvp.test(fold), :) = foldPredictions;
end

% 计算验证 RMSE
isNotMissing = ~isnan(validationPredictions) & ~isnan(response);
validationRMSE = sqrt(nansum(( validationPredictions - response ).^2) / numel(response(isNotMissing) ));
```

```
MLR.m
%% MLR
[b,bint,r,rint,stats] = regress(T_train.',[ones(1700,1) P_train.'])
y_pred=b(1,1)+b(2:21,1).'*P_test
MAPELR=sum(abs(T_test-y_pred)./T_test)/size(T_test,2)
%%
MSELR=sum((T_test-y_pred).^2)/size(T_test,2)
RMSELR=sqrt(MSELR)
MAELR=sum(abs(T_test-y_pred))/size(T_test,2)
RRLR=1-sum((T_test-y_pred).^2)/sum((T_test-sum(T_test)/size(T_test,2)).^2)
```

程序编号	3	程序说明	问题三代码	编写软件	Python

```
# 2021 数学建模-问题 3'
# 部分代码节选，后续将上传笔者代码仓库

# %%
# -*- coding:utf-8 -*-

"""
    Editor: ------ 隐去个人信息 ------
    Date:2021.10.15
"""

# 导入模块
import pandas as pd
import xgboost as xgb
import numpy as np
```

```
import seaborn as sns
from xgboost import plot_importance
import matplotlib.pyplot as plt

from sklearn import metrics
from sklearn.dummy import DummyRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, ExtraTreesRegressor
from sklearn.ensemble import GradientBoostingRegressor, AdaBoostRegressor, BaggingRegressor
from sklearn.model_selection import train_test_split

from plotnine import *
import plotnine


# %%
# 读取数据

Data_train_D = pd.read_excel('../../2021 年 D 题/Molecular_Descriptor.xlsx',sheet_name='training')
Data_test_D = pd.read_excel('../../2021 年 D 题/Molecular_Descriptor.xlsx',sheet_name='test')

# Data_train_E = pd.read_excel('2021 年 D 题/ERα_activity.xlsx',sheet_name='training')
# Data_test_E = pd.read_excel('2021 年 D 题/ERα_activity.xlsx',sheet_name='test')

Data_train_A = pd.read_excel('../../2021 年 D 题/ADMET.xlsx',sheet_name='training')
Data_test_A = pd.read_excel('../../2021 年 D 题/ADMET.xlsx',sheet_name='test')


# %%
# 数据结构整理

Mo_train = Data_train_D.iloc[:, 1:730].as_matrix()
Mo_test = Data_test_D.iloc[:, 1:730].as_matrix()
# ER_train = Data_train_E.iloc[:, 2:].as_matrix()
# ER_test = Data_test_E.iloc[:, 2:].as_matrix()

A_train = Data_train_A.iloc[:, 1].as_matrix()
A_test = Data_test_A.iloc[:, 1].as_matrix()

D_train = Data_train_A.iloc[:, 2].as_matrix()
D_test = Data_test_A.iloc[:, 2].as_matrix()

M_train = Data_train_A.iloc[:, 3].as_matrix()
M_test = Data_test_A.iloc[:, 3].as_matrix()

E_train = Data_train_A.iloc[:, 4].as_matrix()
E_test = Data_test_A.iloc[:, 4].as_matrix()

T_train = Data_train_A.iloc[:, 5].as_matrix()
```

```
T_test = Data_test_A.iloc[:, 5].as_matrix()


# %%
# 准备训练，测试数据集

# for Part Absorption 随机数种子=1
train_x, test_x, train_y, test_y = train_test_split(Mo_train, A_train, test_size=0.2, random_state=1)


# %%
# XGBoost initialize 初始化

dtrain=xgb.DMatrix(train_x,label=train_y)
dtest=xgb.DMatrix(test_x)
watchlist = [(dtrain,'train')]


# %%
# Build Boost model 建模
# 二分-逻辑回归

params={'booster':'gbtree',
        'objective': 'binary:logistic',
        'eval_metric': 'auc',
        'max_depth':5,
        'lambda':10,
        'subsample':0.75,
        'colsample_bytree':0.75,
        'min_child_weight':2,
        'eta': 0.025,
        'seed':0,
        'nthread':8,
        'gamma':0.15,
        'learning_rate' : 0.01}

bst=xgb.train(params,dtrain,num_boost_round=50,evals=watchlist) # 50 tree

# 预测
ypred=bst.predict(dtest)


# %%
# 打印评价指标 - 二分类

y_pred = (ypred >= 0.5)*1
print ('Precesion: %.4f' %metrics.precision_score(test_y,y_pred))
print ('Recall: %.4f' % metrics.recall_score(test_y,y_pred))
print ('F1-score: %.4f' %metrics.f1_score(test_y,y_pred))
```

```
print ('Accuracy: %.4f' % metrics.accuracy_score(test_y,y_pred))
print ('AUC: %.4f' % metrics.roc_auc_score(test_y,ypred))


# %%
# 打印结果

ypred = bst.predict(dtest)
print("测试集每个样本的得分\n",ypred[:51])
ypred_leaf = bst.predict(dtest, pred_leaf=True)
print("测试集每棵树所属的节点数\n",ypred_leaf[:51])
ypred_contribs = bst.predict(dtest, pred_contribs=True)
print("特征的重要性\n",ypred_contribs[:51])


# %%
# function -> feature map -> 配置变量名图

def ceate_feature_map(features):
    outfile = open('xgb.fmap', 'w')
    i = 0
    for feat in features:
        outfile.write('{0}\t{1}\tq\n'.format(i, feat))
        i = i + 1
    outfile.close()

df = (Data_train_D.iloc[0,1:]).index
print(df)

ceate_feature_map(df)


# %%
# 可视化

fig,ax = plt.subplots(figsize=(25,15))

# importance_type -> 关键性指标
temp = xgb.plot_importance(bst, ax=ax,height=0.5, max_num_features=10, importance_type='total_gain',
fmap='xgb.fmap')

plt.title('Important Feature of Absorption (Caco-2)', fontsize=20)
# plt.xlabel("F Score", fontsize=20)
plt.ylabel('Feature Name', fontsize=20)
plt.show()
plt.savefig('../figures/20211015-Absorption.jpg')


# %%
```

```
# 关键性指标打印 -> five types
for importance_type in ('weight', 'gain', 'cover', 'total_gain', 'total_cover'):
    print('%s: ' % importance_type, bst.get_score(importance_type=importance_type))


# %%
# 计算问题结果 Absorb

y_final_results = bst.predict(xgb.DMatrix(Mo_test))

# 打印
print(y_final_results)


# %%
# 储存 numpy 数组

results = np.array([])
for i in y_final_results:
    if i < 0.45840001106262207:
        results = np.append(results, bool(0))
    else:
        results = np.append(results, bool(1))

np.savetxt('../results/results_A',results, delimiter=',')


# %%
# 确定最优阈值

# 创建 ROC 曲线
fpr, tpr, thresholds = metrics.roc_curve(test_y, ypred)

# 绘制 ROC 曲线
df_fpr_tpr = pd.DataFrame({'FPR':fpr, 'TPR':tpr, 'Threshold':thresholds})
df_fpr_tpr.head()


# 计算 G-mean
gmean = np.sqrt(tpr * (1 - fpr))

# 查找最佳阈值
index = np.argmax(gmean)
thresholdOpt = round(thresholds[index], ndigits = 4)
gmeanOpt = round(gmean[index], ndigits = 4)
fprOpt = round(fpr[index], ndigits = 4)
tprOpt = round(tpr[index], ndigits = 4)
print('Best Threshold: {} with G-Mean: {}'.format(thresholdOpt, gmeanOpt))
```

```
print('FPR: {}, TPR: {}'.format(fprOpt, tprOpt))

# 创建数据
plotnine.options.figure_size = (8, 4.8)
(
   ggplot(data = df_fpr_tpr)+
   geom_point(aes(x = 'FPR',
                  y = 'TPR'),
              size = 0.4)+
   # 最佳阈值
   geom_point(aes(x = fprOpt,
                  y = tprOpt),
              color = '#981220',
              size = 4)+
   geom_line(aes(x = 'FPR',
                 y = 'TPR'))+
   geom_text(aes(x = fprOpt,
                 y = tprOpt),
             label = 'Optimal threshold \n for class: {}'.format(thresholdOpt),
             nudge_x = 0.14,
             nudge_y = -0.10,
             size = 10,
             fontstyle = 'italic')+
   labs(title = 'ROC Curve')+
   xlab('False Positive Rate (FPR)')+
   ylab('True Positive Rate (TPR)')+
   theme_minimal()
)
```

程序编号	4	程序说明	问题四代码	编写软件	Python

```
# 数学建模问题四'
# 要添加一个新的标记单元，输入 '# %% [markdown]'

# %%
# 导入模块
import pandas as pd
import xgboost as xgb
import numpy as np
import seaborn as sns
from xgboost import plot_importance
import matplotlib.pyplot as plt
from xgboost import plot_tree

from sklearn import metrics
from sklearn.dummy import DummyRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, ExtraTreesRegressor
```

```
from sklearn.ensemble import GradientBoostingRegressor, AdaBoostRegressor, BaggingRegressor
from sklearn.model_selection import train_test_split

from plotnine import *


# %%
# 数据读取
data = pd.DataFrame(pd.read_excel('/home/yayu/SMJS/2021 年 D 题/MD_17_night.xlsx'))
ER = (pd.DataFrame(pd.read_excel('/home/yayu/SMJS/2021 年 D 题/ER α _activity_includingZ.xlsx'))).iloc[:,3]
data2 = ((pd.DataFrame(pd.read_excel('/home/yayu/SMJS/2021 年 D 题/ADMET_trans.xlsx'))).iloc[:, 1:6])
A, D, M, E, T = data2.iloc[:,0], data2.iloc[:,1], data2.iloc[:,2], data2.iloc[:,3], data2.iloc[:,4]

miu = 6.5
delta = 1.4


# %%
# 分类指标计算

# A1 + 0.5 * B1 + 1
A1 = (A + D + E + T + M) * 0.1
B1 = ER

new_set = A1 + B1*0.5 + 1

# indicators
indicators = np.array([])
for i in new_set:
    # level best 1.72125
    if i > 1.72125:
        indicators = np.append(indicators, 2)
    # level normal 1.378
    elif i>1.378:
        indicators = np.append(indicators, 1)
    # level bad else
    else:
        indicators = np.append(indicators, 0)

np.savetxt('myfile.txt',indicators)


# %%
Target_train = indicators
Matrix_train = data.iloc[:,1:]
# sklearn 库 split()函数切分 test 矩阵数据集

# 随机数种子=1
```

```
train_x, test_x, train_y, test_y = train_test_split(Matrix_train, Target_train, test_size=0.2, random_state=1)

# XGBoost initialize
dtrain=xgb.DMatrix(train_x,label=train_y)
dtest=xgb.DMatrix(test_x)
watchlist = [(dtrain,'train')]


# %%
# Build Boost model 建模
# 多分类

params={'learning_rate': 0.1,
        'max_depth': 5,
        'num_boost_round':20,
        'objective': 'multi:softmax',
        'random_state': 27,
        'silent':0,
        'num_class':3

}

bst=xgb.train(params,dtrain,num_boost_round=50,evals=watchlist) # 50 tree


# %%
# 预测
ypred=bst.predict(dtest)

bst.save_model('../models/testXGboostClass.model') # 保存训练模型


# %%
# 模型评估

print("Accuracy:",round(metrics.accuracy_score(y_true=test_y, y_pred=ypred), 4))

# 特征重要性
xgb.plot_importance(bst)


# %%
# 可视化
# X 轴 特征值
# Y 轴 评价标准

plt.scatter(Matrix_train.iloc[:,1], Target_train)
```

```
# %%
# 设置预测数据范围

# 4 个变量
plt.figure(figsize=(10,10))
data_test = pd.DataFrame(pd.read_excel('/home/yayu/SMJS/2021 年 D 题/test_class.xlsx'))

test1, test2, test3, test4 = data_test.iloc[:,4], data_test.iloc[:,5], data_test.iloc[:,6], data_test.iloc[:,7]

set1, set2, set3, set4 = np.array([]),  np.array([]),  np.array([]),  np.array([])

num = 0
for i in indicators:
    if i == 2:
        plt.scatter(num,test1[num],c='r', linewidths=0.1)
        plt.scatter(num,test2[num],c='y', linewidths=0.1)
        plt.scatter(num,test3[num],c='g', linewidths=0.1)
        plt.scatter(num,test4[num],c='b', linewidths=0.1)
        set1 = np.append(set1, test1[num])
        set2 = np.append(set2, test2[num])
        set3 = np.append(set3, test3[num])
        set4 = np.append(set4, test4[num])



    elif i ==1:
        # plt.scatter(num,test1[num],c='b' ,linewidths=0.1)
        pass
    else:
        # plt.scatter(num,test1[num],c='k', linewidths=0.1)
        pass
    num += 1
plt.legend(loc='best')
plt.show()


# %%
plt.figure(figsize=(10,5), dpi=200)
plt.boxplot([set1, set2, set3, set4],vert=False,showmeans=True)
plt.show()
```