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	武汉大学
参赛队号	21104860088
队员姓名	1.陈苗苗
	2.孙冉
	3.王继莲

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：

在研发治疗乳腺癌药物的过程中，能拮抗 ERα活性的化合物是治疗乳腺癌的重要候选药物，同时也要考虑到化合物在人体内具备良好的药代动力学性质和安全性(ADMET 性质)，如果吸收性能、代谢速度、毒副作用等性质不佳，依然很难成为药物。本文对给定的 1974 个化合物的分子描述符、生物活性以及 ADMET 性质进行处理分析，探寻对生物活性有重要影响的分子描述符，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，并基于两个模型构建目标优化模型，找出具体的分子描述符范围。文章综合运用了**内置随机森林重要性、基于排列的重要性、SHapley Additive exPlanation 特征重要性排序、决策树、逻辑回归、Light Gradient Boosting Machine（LightGBM）、相关性分析、证据权重 WOE 和信息值 IV 筛选、过采样、机器学习、XGBoost 分类、RandomForest 分类、粒子群算法、主要目标法**等经典机器学习算法和分析方法对相关问题进行量化分析和数学建模，使用了 Python、MATLAB 等软件实现模型并得到问题答案。

针对问题一（变量选择），通过观察发现，数据中存在**稀有变量和异常值**的现象，首先，剔除 270 个稀有变量及使用均值法填充少量异常值，然后，通过 **Spearman 相关性**分析发现 729 个分子描述符之间存在一些相关性很高的变量，对相关性较高的 213 个变量以及与生物活性相关性较低的 50 个变量进行初步筛选，最终得到 196 个分子描述符。随后，构建**随机森林模型**，利用**基于内置随机森林重要性、基于排列的重要性、基于 SHapley Additive exPlanation 的重要性**三种特征重要性计算方法进行特征重要性排序，筛选出前 20 个对生物活性最具有显著影响的分子描述符，发现 **MDEC-23**、**LipoaffinityIndex**、**maxHsOH** 在三种特征重要性计算算法下均排名前三，随后选择可解释性的 **SHAP 算法**分析前 20 个分子描述符对生物活性的正负影响程度。

针对问题二（生物活性定值预测），首先，本文构建了基于**决策树、逻辑回归、线性回归、Light Gradient Boosting Machine（LightGBM）**等十二类算法的 ERα生物活性的定量回归预测模型，为了更好地筛选出重要的描述符，在次选择的变量为问题一中经过预处理和相关性筛选后的变量，随后采用 **MSE**、**RMSE** 等指标对各个模型的性能进行评估。结果发现，基于 LightGBM 算法的生物活性定值回归预测模型表现效果最好，MSE 值为最低 **0.4424**。随后，本文计算基于 LightGBM 算法的分子描述符的 SHAP 值，并与问题一得

到的前 20 个对生物活性最具有显著影响的分子描述符进行对比，选择出了交叉的 15 个分析描述符作为特征，对模型的参数（如 max_depth 和 num_leaves）进行调整，得到性能最优的生物活性定值回归预测模型，最后，对文件“ERα_activity.xlsx”的 test 表中的 50 个化合物进行 pIC50 值预测，并通过 pIC50 值计算对应的 IC50 值。

针对问题三（分类模型构建和预测），首先，基于**证据权重 WOE 和信息值 IV 方法**，对影响化合物的 Caco-2、CYP3A4、hERG、HOB、MN 五种 ADMET 性质的分子描述符进行筛选，确定用于预测不同 ADMET 性质的变量类型。其次，对五种 ADMET 性质的分类变量数据分布进行分析，采用**过采样方法均衡数据样本**。再次，**构建 13 种分类模型，包括 11 种机器学习模型和 2 种深度学习模型（LSTM、CNN）**，通过对各个模型准确率、精度、召回率、F1 值、ROC 曲线、AUC 值及对数损失等指标的评价和比较，**确定预测 Caco-2、CYP3A4、hERG 三种 ADMET 性质的最佳模型为 XGBoost 分类模型，预测 HOB、MN 两种 ADMET 性质的最佳模型为随机森林（RandomForest）分类模型**。然后，为了预测 test 集合中的 ADMET 性质，**一方面，评估了最佳模型的泛化能力，结果表明各个模型泛化能力较强**，在测试集上的学习能力分数都达到了 0.9 以上，其中，预测 MN 的 RanomForest 分类模型和预测 CYP3A4 的 XGBoost 分类模型在测试集上学习能力分数超过了 0.96；另一方面，为了提高模型预测的准确率，基于十折交叉验证方法，**在训练集上对获得的最佳模型进行参数调优**，以获得最优模型，并基于最优参数下的最优模型预测 ADMET 性质，结果表明在参数调优下，预测 Caco-2 的 XGBoost 模型准确率最高可达到 **93.9%**；预测 CYP3A4 的 XGBoost 模型准确率最高可达到 **96.8%**；预测 hERG 的 XGBoost 模型准确率最高可达到 **92.6%**；预测 HOB 的 RandomForest 模型准确率最高可达到 **92%**；预测 MN 的 RandomForest 模型准确率最高可达到 **97.8%**。最后，本文对 5 个分类模型的**特征重要性排序和重要变量的关系进行简要的可视化分析**，结果表明不同分类模型的重要特征具有明显差异，不同重要变量的组合对分类结果的影响也有明显差异。

针对问题四（优化选取分子描述符并估计范围），本文构建了以提高抑制 Erα的生物活性和提高 ADMET 性质为目标的**多目标优化模型**，根据前文进行数据清理得到的主要分子描述符，基于问题二、三所构造的回归模型和分类模型构造出符合题目要求的多目标优化模型，然后通过**粒子群算法**权衡两个目标函数之间的关系，求解 Pareto 解集，之后再通过**主要目标法**将模型再次求解验证答案的有效性，并从中选出部分分子描述符展示目标函数最优的取值范围，并且通过散点图可视化部分分子描述符的重要影响和取值范围。

关键词：生物活性分析；逻辑回归；分类预测；随机森林；XGBoost；机器学习；相关性分析；粒子群算法；多目标优化

目录

一、前言 5
1.1 问题背景 5
1.2 问题重述 5
1.3 基于 VOSviewer 的关键词共现时序分析 5
二、模型假设 7
2.1 模型基本假设 7
2.2 模型符号说明 7
三、技术路线图 8
四、问题一：模型的建立与求解 9
4.1 问题分析 9
4.2 数据预处理 9
4.2.1 去除稀有变量 9
4.2.1 去除异常值 10
4.3.1 描述符之间的相关性分析 11
4.3.2 描述符与药物活性之间的相关性 12
4.4 变量方法的构建与评估 13
4.4.1 变量方法调研 13
4.4.2 基于随机森林的变量选择模型 13
4.5 特征重要性排序方法和结果 13
4.5.1 基于排列（Permutation-based）的特征重要性 13
4.5.2 基于 SHAP 的药物优化特征重要性排序 14
4.5.3 特征重要性排序结果 14
4.5 模型小结 15
五、问题二：模型的建立与求解 17
5.1 问题分析 17
5.1 定值回归模型的构建和特征描述符的选择 17
5.1.1 回归方法调研 17
5.1.2 基于 LightGBM 算法的药物优化回归模型的构建 17
5.1.3 多种回归模型的评估 19
5.2 LightGBM 回归预测模型参数调优 21
5.2.1 参数调优: max_depth 和 num_leaves 21
5.2.2 参数调优: min_data_in_leaf 和 max_bin 21
5.2.3 参数调优: feature_fraction、bagging_fraction 22
5.3 药物活性的回归模型的预测结果分析 23
5.4 模型小结 23
六、问题三：模型的建立与求解 24
6.1 问题分析和技术路线图 24
6.2 基于证据权重 WOE 和信息值 IV 的变量筛选 25
6.2.1 WOE 与 IV 的计算 25
6.2.2 变量筛选 26
6.3 模型建立 28
6.3.1 11 种机器学习模型建立 28

6.3.2 LSTM、CNN 两种深度学习模型建立 28
6.4 模型筛选 30
6.4.1 筛选模型的评价指标 30
6.4.2 ADMET 性质数据预处理-过采样及样本划分 31
6.4.3 筛选构建 Caco-2 的分类预测模型 32
6.4.4 筛选构建 CYP3A4 的分类预测模型 35
6.4.5 筛选构建 hERG 的分类预测模型 37
6.4.6 筛选构建 HOB 的分类预测模型 39
6.4.7 筛选构建 MN 的分类预测模型 41
6.5 基于 XGBoost 和随机森林分类模型的不同 ADMET 性质模型求解 43
6.5.1 模型参数调优 43
6.5.2 基于最优 XGBoost 和随机森林分类模型的 ADMET 性质求解 46
6.6 特征重要性排序和变量关系可视化分析 47
6.7 模型小结 48
七、问题四：模型的建立与求解 50
7.1 问题分析 50
7.2 数据预处理 50
7.3 模型建立与求解 51
7.3.1 基于粒子群算法的多目标优化问题 51
7.3.2 基于主要目标法的优化问题 53
7.3.3 基于粒子群算法模型的求解结果 54
7.3.4 基于主要目标法的求解结果 54
7.3.5 结果分析 55
7.4 模型小结 57
八、 参考文献 58
九、 附录 59

一、前言

1.1 问题背景

乳腺癌发病多见于女性且近年来发病率在年轻人群体中所占有的比率逐渐增大，致死率较高，对于乳腺癌的发病机制目前的研究并未完全透彻，但发现其与雌激素受体密切相关，实验表明 ERα在乳腺癌治疗中起着十分重要的作用，因此在临床治疗时，通常选用能拮抗 ERα活性的化合物作为治疗乳腺癌的候选药物，如他莫昔芬可以显著提高乳腺癌患者的生存几率。但是由于药物研发通常需要极高的时间、精力成本，为做好乳腺癌药物的研发工作，通常会建立化合物的定量结构-活性关系（Quantitative Structure-Activity Relationship, QSAR）模型,即利用计算和统计的方法定量研究化合物的二维结构、三维结构等结构特征与化合物的药性、活性等生活效应特征之间的关系，该方法也是药物分子学设计中重要的方法基础和理论基础，能够指导药物分子结构的优化，通过该方法计算所得到的参数值可以指导生物活性预测、药物分子设计等[1]。

良好的乳腺癌生物活性是一个化合物成为候选药物必备的条件之一，除此之外，该化合物还需要具备稳定性、安全性、可代谢、低毒性等特征，以便能够适应人体内部化学生物反应，化合物对人体的药代动力学性质通常是由 ADMET 性质描述（Absorption 吸收、Distribution 分布、Metabolism 代谢、Excretion 排泄、Toxicity 毒性），好的性质是该化合物成为候选药的另一个条件。

本题的命题宗旨是利用相关的数据建立影响生物活性的定量预测模型及构建 5 种 ADMET 性质的分类预测模型，两个模型的精度和泛化能力越高越好，此外，还需要找出重要的分子描述符及其取值范围或值从而为化合物同时优化 ERα拮抗剂的生物活性和 ADMET 性质提供相应指导。

1.2 问题重述

基于前述研究背景，该题目共提供了四个附件，即“ERα_activity.xlsx”、“Molecular_Descriptor.xlsx”、“ADMET.xlsx”和“分子描述符含义解释.xlsx”，基于四个附件内容，拟要解决的研究问题如下：

（1） **寻找影响生物活性的重要变量**：依据变量对生物活性影响的重要程度对 729 个分子描述符进行变量选择并进行排序，给出前 20 个对生物活性具有显著影响的分子描述符。

（2） **构建影响生物活性的定量预测模型**：选择不超过 20 个分子描述符变量，构建化合物对 ERα生物活性的定量预测回归模型，并预测“ERα_activity.xlsx”test 表中 50 个化合物的 IC50 值和对应的 pIC50 值。

（3） **构建 5 种 ADMET 性质的分类预测模型**：将“Molecular_Descriptor.xlsx”提供的 729 个分子描述符视为自变量，“ADMET.xlsx”的五种 ADMET 性质视为因变量，分别构建化合物的 Caco-2、CYP3A4、hERG、HOB、MN 的二分类预测模型，并对“ADMET.xlsx”的 test 表中的 ADMET 性质进行预测。

（4） **确定分子描述符及其取值范围来获得更好的生物活性和更好的 ADMET 性质**：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）。

1.3 基于 VOSviewer 的关键词共现时序分析

为了使研究方向更加精确，本文通过关键词共现来揭示有关药物的生物活性和ADMET优化相关研究的关键内容，以CNKI数据库为数据源，通过主题词“抗肿瘤活性”和“ADMET”进行检索，设置检索时间为2006-2021年，借助可视化分析工具VOSviewer对文章的关键词进行共现与聚类，从关键词的角度对药物的生物活性和ADMET优化的研究内容进行初探。VOSviewer作为知识图谱软件，能可视化揭示文献数据之间的关系。最终的关键词共现时序分析结果见图1.1和图1.2。关键词出现的频次越多，图中的圈越大。目前以及有不少有关抗肿瘤药物活性的研究，包括足校关系模型、有效成分提取、成分分析、分离鉴定、正交试验、响应面优化发等等，涉及到药物动力学、药代动力学、药效学的相关理论研究。与ADMET有关的研究则更多的应用于药物设计和药物发现，与分子对接、虚拟筛选（利用小分子化合物与药物靶标间的分子对接运算，快速地从大量分子中筛选出具有成药性的活性化合物）等有关的研究较为丰富。

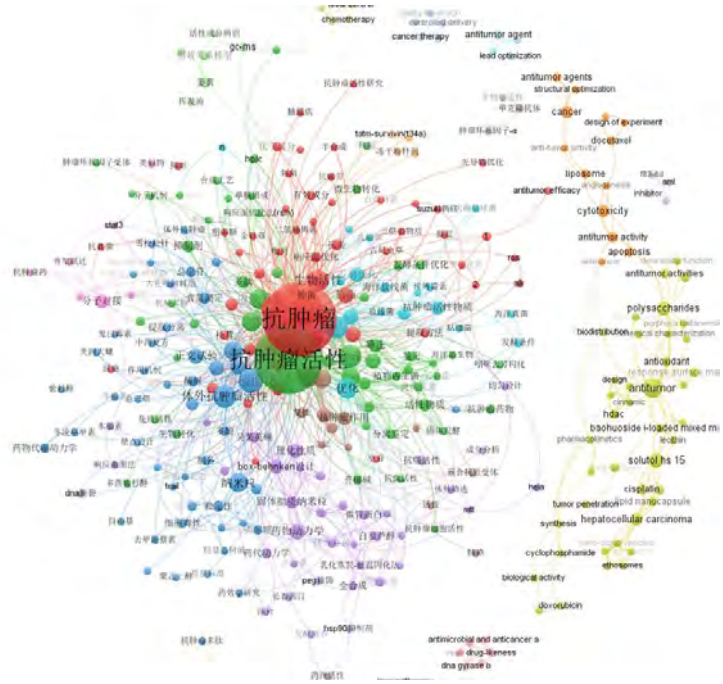


图1.1 “抗肿瘤活性”的关键词共现分析图

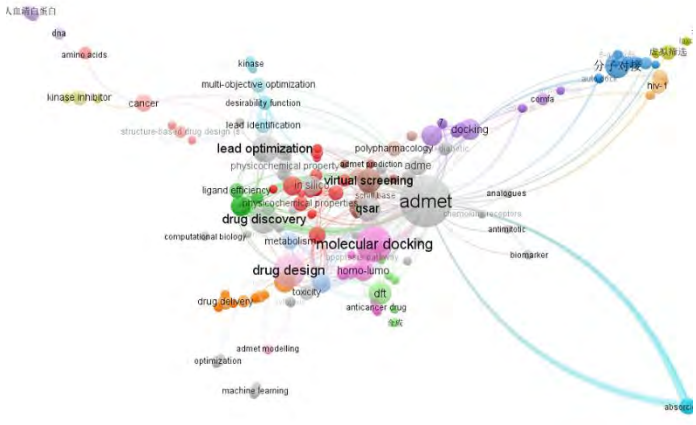


图1.2 “ADMET”的关键词共现分析图

二、模型假设

2.1 模型基本假设

根据化合药物研发情况和本题所给出的条件，本文作出如下假设：

(1) 所给数据中某一分子描述符在所有所给化合物中的值都为 0 时，即这一分子描述符对活性影响并不大。
(2) 除了所给的分子描述符之外，其他存在的因素对生物活性和 ADMET 性质的影响可以忽略。
(3) 分子描述符的所给值默认为连续变量。

2.2 模型符号说明

本文所涉及的模型符号说明如表 2.1 所示

表 2.1 模型符号说明

序号	符号	说明
1	ρ	Spearman 相关系数
2	$D_{1/2}^{R}$	差异度
3	V	信息增益
4	L	损失函数
5	Ω	正则项
6	Obj	目标函数
7	W	权重函数
8	σ	Sigmoid 激活函数
9	p_i	概率
10	“Caco-2”	Caco-2 性质对应的值
11	Z	可行空间
12	v_i	速度
13	$\boldsymbol{present}$	当前位置

三、技术路线图

本文的研究技术路线图如图 3.1 所示：

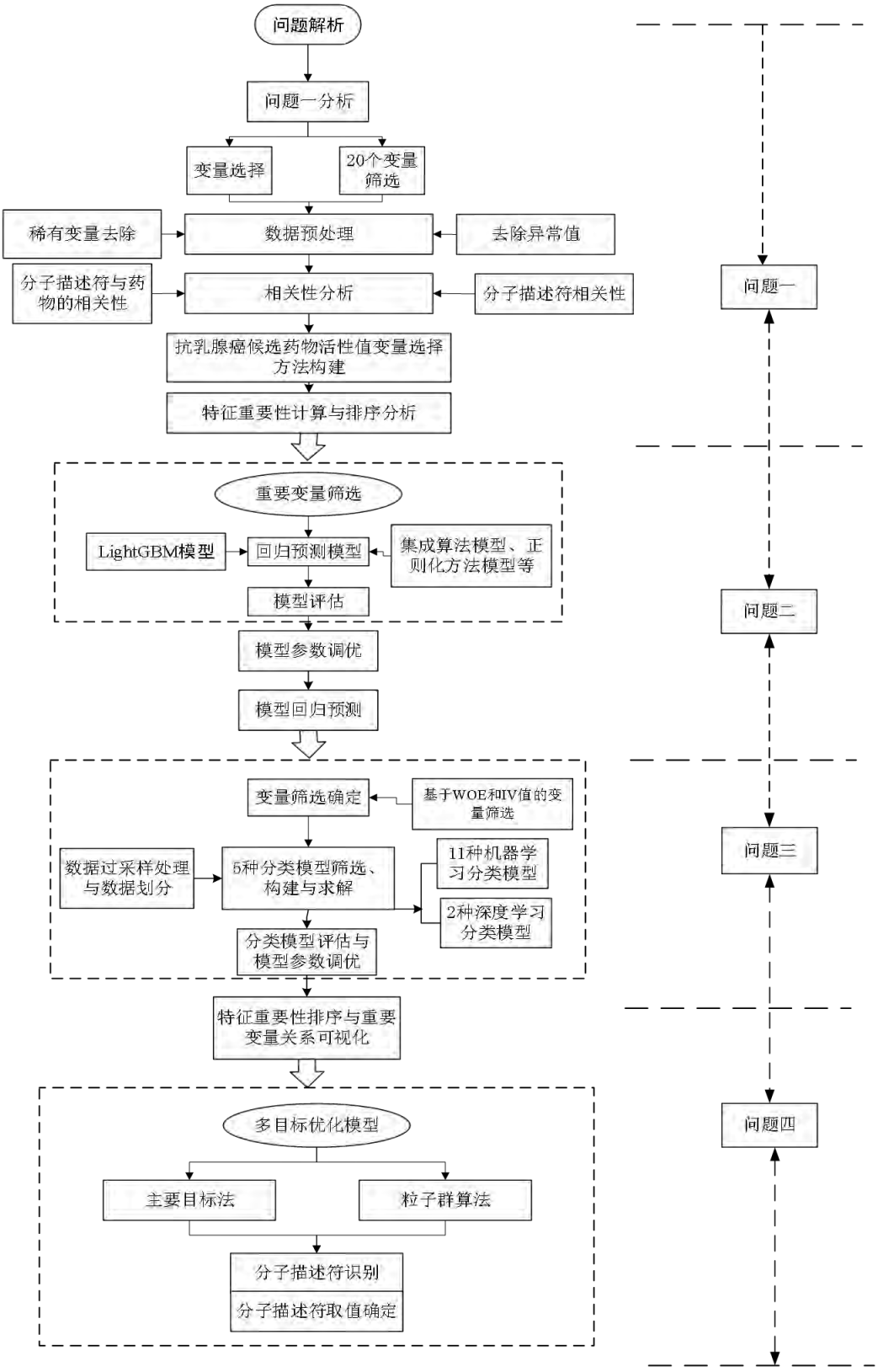


图 3.1 技术路线图

四、问题一：模型的建立与求解

4.1 问题分析

根据题目要求，生物活性的影响因素主要是从 729 个分子描述符中进行选择。拟从以下三个步骤解决问题一：(1)通过对原始数据进行多个变量之间的描述性统计，发现数据中存在大量的稀有变量，因此首先针对数据缺陷问题进行预处理;(2)分析各因变量之间的相关性及其对生物活性的影响，并采用**相关系数法**针对训练集中 1974 个化合物的 729 个分子描述符信息进行初步变量选择。(3)对所选取的变量构建随机森林模型，利用基**于内置随机森林重要性、基于排列的重要性、SHapley Additive exPlanation** 三种特征重要性计算方法进行特征重要性排序，筛选出前 20 个对生物活性最具有显著影响的分子描述符。

确定问题一的总体思路如图 4.1 所示。

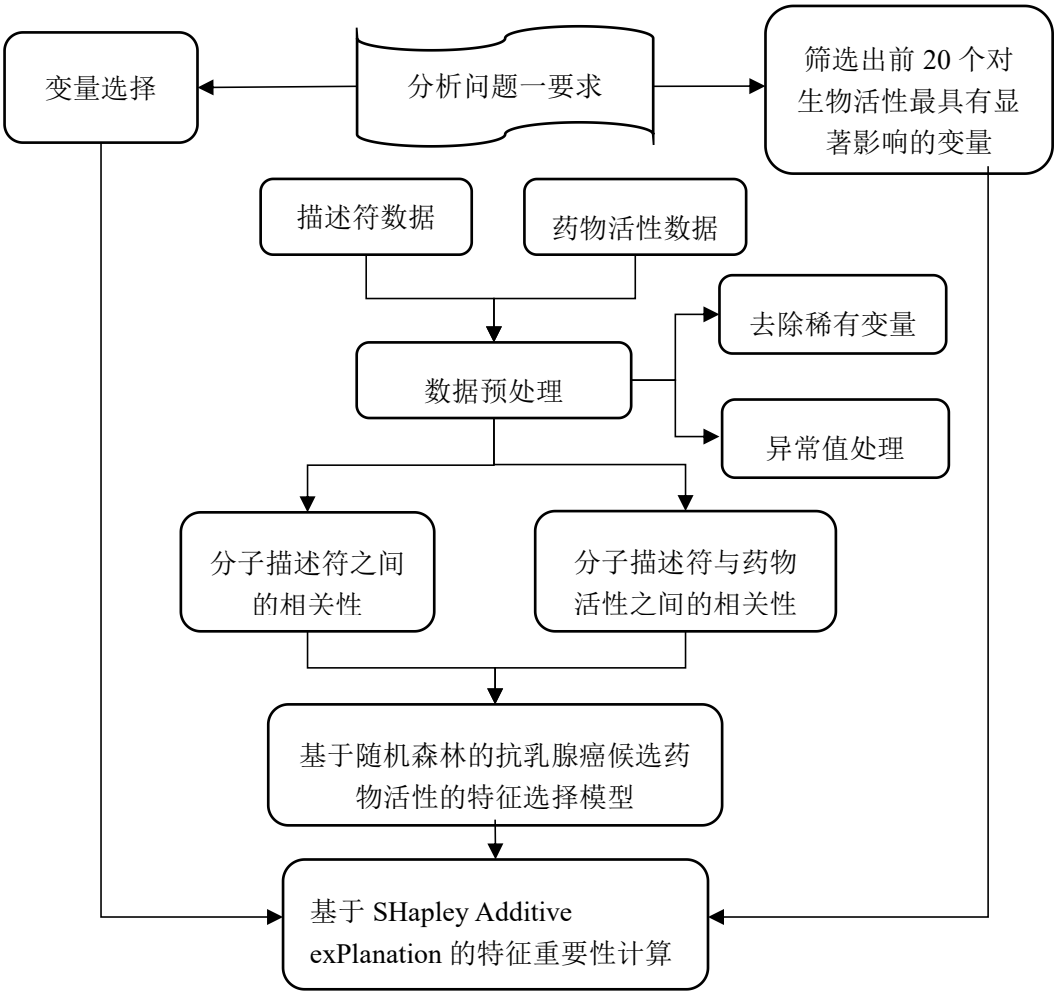


图 4.1 问题一的解题思路

4.2 数据预处理

4.2.1 去除稀有变量

对文件 Molecular_Descriptor.xlsx 中所给的数据进行基本的统计分析，得到表 4.1 所示。可以发现分子描述符的数据分为分类变量、连续型变量，其中包括一些数据值全部为 0 的变量，尽管数值为 0 也具有实际意义，但预测模型并不能识别其意义，同时这些变量会被

认为是冗余特征，从而影响模型的精度。

表 4.1 部分分子描述符的统计信息列表

分子描述符	mean	std	min	0.25	0.5	0.75	max
pIC50	6.586186	1.423052	2.456	5.38225	6.581	7.5685	10.337
nAcid	0.108409	0.3479	0	0	0	0	4
ALogP	1.110164	1.43425	-23.105	0.3763	1.17095	1.9481	5.1817
naAromAtom	15.44681	5.155854	0	12	16	18	30
nAromBond	16.18946	5.635271	0	12	18	18	34
nB	0	0	0	0	0	0	0
nC	22.6079	6.631359	7	17	22	28	95
nN	1.508612	1.886457	0	1	1	2	46
nS	0.307497	0.562536	0	0	0	1	6
nP	0.001013	0.031822	0	0	0	0	1
nCl	0.101317	0.36283	0	0	0	0	6
nBr	0.061297	0.254292	0	0	0	0	3

对照分子描述符含义解释掌握分子描述符的含义。查阅相关文献，为了后续数据的处理，对 Molecular_Descriptor.xlsx 中的数据进行简单处理。保留对数据分析处理和数据建模有价值的数据信息，剔除稀有变量 270 个，剔除的部分分子描述符如下表 4.2 所示。

表 4.2 部分筛选掉的描述符信息列表

编号	分子描述符	含义
1	nBondsQ	四重键数
2	nssGeH2	原子级电子态计数: -GeH2-
3	nsssGeH	原子级电子态计数: >GeH-
4	nssssGe	原子级电子态计数: >Ge<
5	nHmisc	原子级电子态计数: H bonded to B, Si, P, Ge, As, Se, Sn or Pb
6	nsLi	原子级电子态计数: -Li
7	nssBe	原子级电子态计数: -Be-
8	nssssBem	原子级电子态计数: >Be<-2
9	nsBH2	原子级电子态计数: -BH2
10	nssSiH2	原子级电子态计数: -SiH2-

4.2.1 去除异常值

本文采用基于箱线图法对异常值进行识别，以分子描述符 ALogp2 为例，在第 1563 个样本中出现了异常值 517.4294，如图 4.2 所示，对于这些异常值，使用该位点其它数值平均值代替。

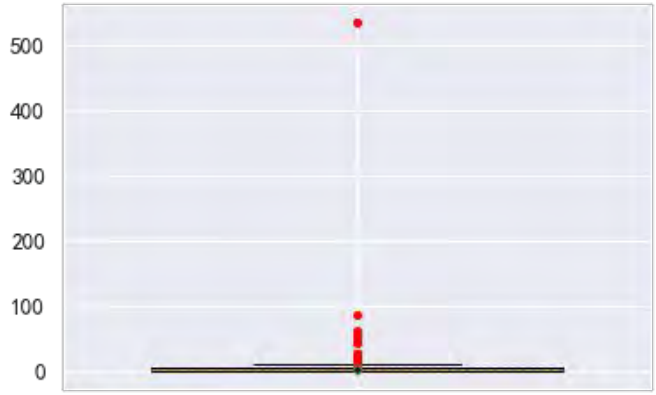


图 4.2 分子描述符 ALogp2 的箱型图

4.3 变量选择

4.3.1 描述符之间的相关性分析

在进行变量的相关性检验之前，本文利用 SPSS 对所有描述符数据进行正太性检验，结果表明大部分数据并不服从正太性检验，因此本文选择更适合处理非正太分布数据的 Spearman 模型进行相关性检验，其公式如（4-1）所示。

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \qquad (4\text{-}1)$$

其中，x,y 为两个待分析变量取值，$\bar{x}, \bar{y}$为两个变量的平均值。相关系数的绝对值越大，则代表两个变量之间的相关性越强。部分变量的相关性如图 4.3 所示，其中，对角线变量表示自相关，其他位置的变量表示互相之间的相关性，颜色越深，接近于红色表明相关性越强，反之变量之间的相关性越弱。相关性分析结果表明部分变量之间存在很强的相关性，而冗余特征会损害模型的可解释性，因此需要剔除相关性高的 213 个特征。

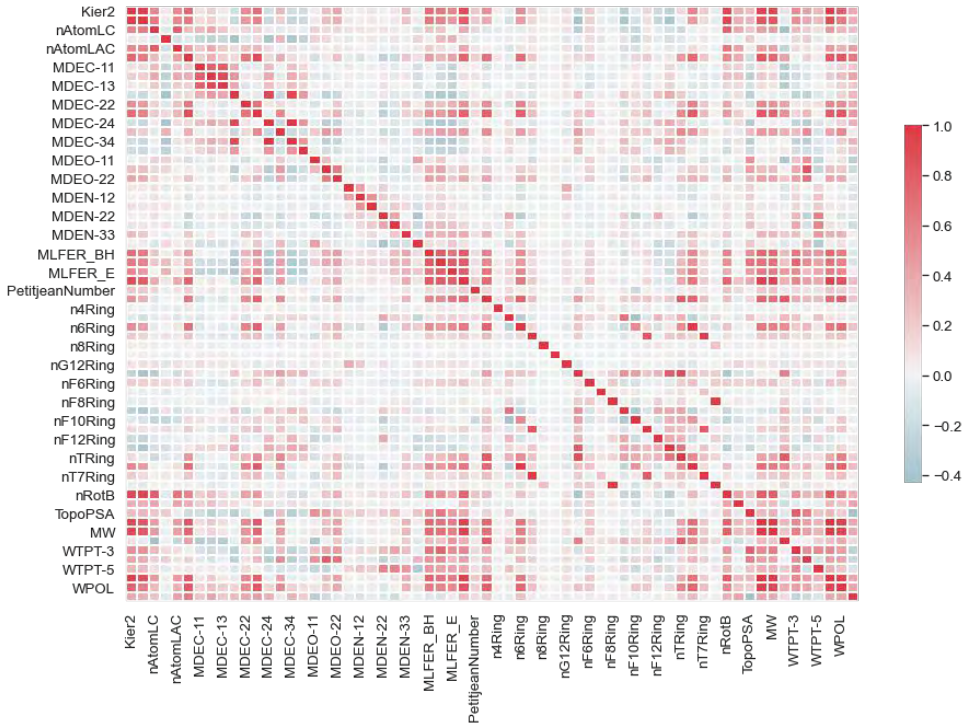


图 4.3 部分变量之间的相关性系数

通过相关性系数筛选出了相关性大于 0.9 的变量，其中部分变量之间的相关性结果如图 4.4 所示，从图中可以看出，这些变量之间呈现高度相关的特征。

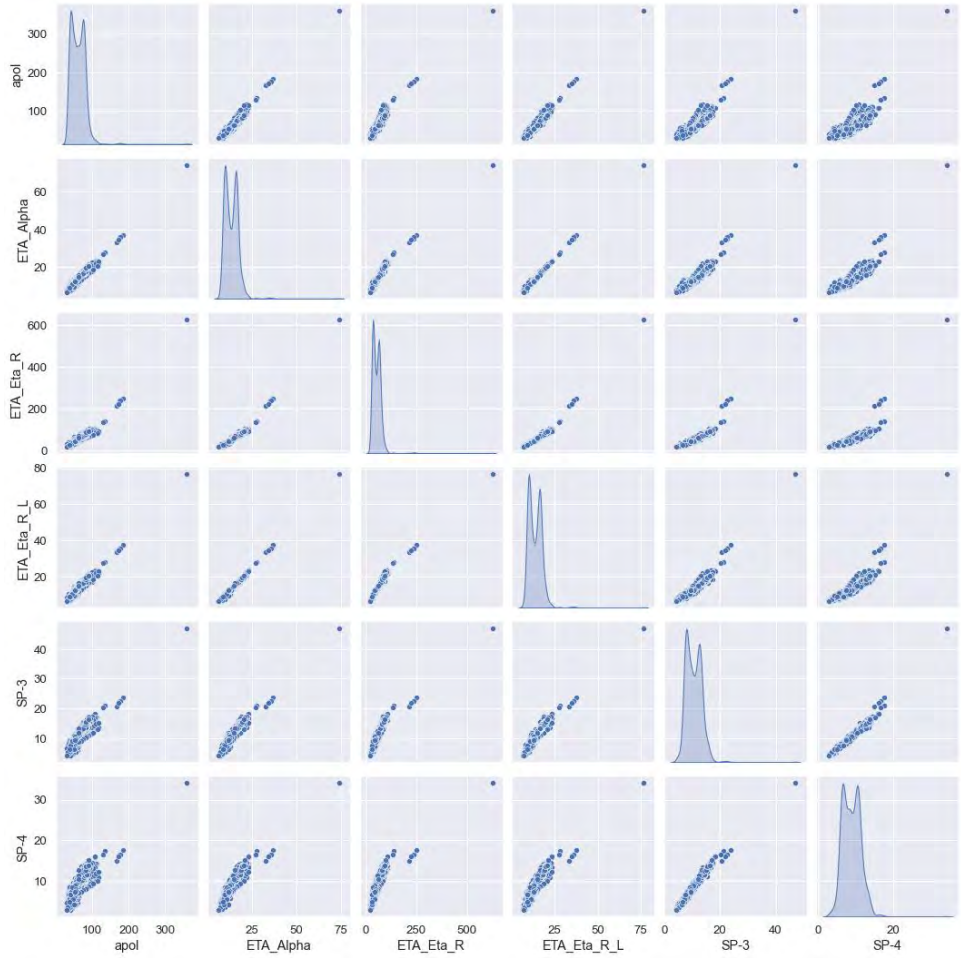


图 4.4 相关性高的部分变量的相关性关联图

4.3.2 描述符与药物活性之间的相关性

为了证明分子描述符与药物活性之间具有相关性，对药物活性和分子描述符之间的相关性进行计算，筛选掉相关性较低的 50 个变量，剔除的部分分子描述符如下表 4.3 所示。

表 4. 3 描述符与药物活性之间的相关性系数

编号	分子描述符	含义
1	VCH-3	Valence chain, order 3
2	VCH-4	Valence chain, order 4
3	VPC-6	Valence path cluster, order 6
4	SP-0	Simple path, order 0
5	SP-1	Simple path, order 1
6	SP-2	Simple path, order 2
7	VP-7	Valence path, order 7
8	ndCH2	Count of atom-type E-State: =CH2
9	ntCH	Count of atom-type E-State: #CH
10	ndsCH	Count of atom-type E-State: =CH-

4.4 变量方法的构建与评估

4.4.1 变量方法调研

变量选择是指拟合线性和非线性的多元回归方程进行自变量选择、拟合判别函数的变量选择等。针对问题一的要求，需要对 1974 个化合物的 729 个分子描述符进行变量选择。目前主流的变量方法有子集选择、收缩方法、维数缩减。对这些变量选择方法的特点和优缺点进行调研，得到结果如下表 4.4 所示。

表 4. 4 回归分析方法比较

	方法描述	优缺点
最优子集选择	将所有的特征组合进行建模，然后根据 AIC、BIC 等准则选择最优的模型，如向前选择和向后选择	适用于小数据量、简单的关系，但经常表现为高方差，因此不容易降低全模型的预测误差
压缩系数法	基于惩罚项变量选择方法，主要指岭回归和 Lasso 回归	用于强相关的自变量时，在进行参数估计时，会导致解不可逆，十分不稳定
降维法	将变量进行变换后的新变量进行降维，主要是指主成分回归和偏最小二乘法	对特征值的分解存在局限性
树结构的方法	如随机森林模型，其本身可用于预测的模型，但在预测过程中，可以对变量重要性进行排序，然后通过这种排序来进行变量筛选	可以用统一的方法处理数值型变量和分类型变量

4.4.2 基于随机森林的变量选择模型

本文采用 Gini 指数作为分子描述符的重要性评判指标，针对一棵树中的每个节点k，计算其 Gini 指数，如公式（4-2）所示。

$$G_k = 2\hat{p}_k(1-\hat{p}_k) \tag{4-2}$$

其中，$\hat{p}_k$代表样本在节点k属于任何一类的概率估计值，一个节点的重要性程度由节点分裂前后 Gini 指数的变化量来确定：

$$I_{\Delta k} = G_k - G_{k1} - G_{k2} \tag{4-3}$$

其中，G_{k1}和G_{k2}分别表示节点k产生的子节点，针对每棵树进行递归，最终随机抽样样本和变量，产生包含 T 棵数的森林，如果变量X_i在第t棵树中出现 N 次，则变量X_i的重要性为：

$$I_{it} = \frac{1}{n}\sum_{t=1}^{T}\sum_{j=1}^{N} I_{\Delta j} \tag{4-4}$$

4.5 特征重要性排序方法和结果

4.5.1 基于排列（Permutation-based）的特征重要性

除了内置随机森林重要性（Gini 指数）计算，还可以采用基于排列的特征重要性计算公式。对于样本数据 D 中的特征 j，随机打乱数据集 D 中的第 j 列取值，并将新的数据集

命名为$\widehat{D}_{k,j}$，计算模型在数据集$\widehat{D}_{k,j}$上的分数$s_{k,j}$，则计算特征f_i的重要性I_j如以公式（4-5）所示：

$$I_j = s - \frac{1}{K}\sum_{k=1}^{K} s_{k,j} \tag{4-5}$$

4.5.2 基于 SHAP 的药物优化特征重要性排序

特征重要性是一种为预测模型的输入特征进行评分的方法，可以揭示进行预测时每个特征的相对重要性。SHapley Additive exPlanation（SHAP）属于模型事后解释的方法，可以增强复杂机器学习模型的可解释性。因此，可以采用 SHAP 算法对模型中对药物活性优化的影响因素进行解释分析。

SHAP 值来源于博弈论中的 Shapley value，主要是用于评估每个特征对模型预测的贡献值，其基本原理是计算每个特征对模型的边际贡献，然后计算该特征在所有特征序列中不同的边界贡献，最后该特征所有边际贡献的均值即为 SHAP 值。

假设模型基准分（所有样本的目标变量的均值）为y_{base}，第i个样本为x_i，第i个个样本的第j个特征为x_{ij}，特征的边际共现为ms_{ij}，边的权重为w_k，模型对该样本的预测值为y_i，则第 i 个样本的第 1 个特征的 SHAP 值$f(x_{i1})$如(4-6)所示，同时 SHAP 值要服从公式(4-7)。

$$f(x_{ij}) = \sum_{k=1}^{n} ms_{i1} w_k \tag{4-6}$$

$$y_i = y_{base} + \sum_{s=1}^{n} f(x_{is}) \tag{4-7}$$

4.5.3 特征重要性排序结果

本文对比分析三种不同的特征重要性算法的排序结果，如表 4.5 所示，输入模型的变量为经过数据预处理得到的 196 个分子描述符，从表中发现，分子描述符 MDEC-23、LipoaffinityIndex、maxHsOH 在三种特征重要性计算算法下均排名前三，尽管之后的分子描述符的特征重要性排序有所不同，但对定值预测模型的影响较小。

表4. 5 三种特征重要性排序算法的结果对比

分子描述符	特征重要性（GINI）	分子描述符	特征重要性(Permutation)	分子描述符	特征重要性(SHAP)
MDEC-23	0.171	MDEC-23	0.126	MDEC-23	0.312698075
LipoaffinityIndex	0.077	LipoaffinityIndex	0.05	LipoaffinityIndex	0.178193013
maxHsOH	0.05	maxHsOH	0.04	maxHsOH	0.116011042
minsssN	0.044	minHsOH	0.024	BCUTc-1l	0.09790052
minHBint5	0.038	C1SP2	0.021	C1SP2	0.093620064
BCUTc-1l	0.038	VC-5	0.017	minHBint5	0.082440564
C1SP2	0.032	BCUTc-1l	0.017	minsssN	0.073858329
minHsOH	0.03	nHBAcc	0.016	minHsOH	0.070860766
nHBAcc	0.02	minHBint5	0.015	nHBAcc	0.066484049

minsOH	0.014	ATSc3	0.01	VC-5	0.037619015
ATSc3	0.014	minsssN	0.008	ATSc3	0.031815638
VC-5	0.013	minsOH	0.007	minsOH	0.026887697
SHBint10	0.01	TopoPSA	0.005	XLogP	0.023957633
MLFER_A	0.01	MLFER_A	0.005	MLFER_A	0.023897194
XLogP	0.009	MDEC-33	0.004	ndssC	0.02354074
maxHBa	0.009	VCH-5	0.004	CrippenLogP	0.023160492
MDEO-12	0.008	MDEC-22	0.004	MDEO-12	0.019767238
hmin	0.008	CrippenLogP	0.003	mindssC	0.019740989
CrippenLogP	0.008	MDEO-12	0.003	TopoPSA	0.018172734
TopoPSA	0.008	gmin	0.003	maxHBint5	0.01770153

由于传统的特征重要性排序无法反映出特征如何影响预测结果，而SHAP能反映出每一个样本中的特征的影响力，同时能展示出影响的正负性。从随机森林模型和SHapley Additive exPlanation算法中得到的特征重要性排序结果如图4.5所示，从图中可以发现，MDEC-23对药物的生物活性影响最大，随着MDEC-23（所有二级和三级碳之间的分子距离边缘）、LipoaffinityIndex（脂亲和指数）、maxHsOH（最大原子类型H E态：-OH）、minHBint5（路径长度为5的潜在氢键强度的最小电子态描述符）、minsssN（最小原子类型电子态：>N-）等分子描述符值的增加，生物活性越大的可能性增加。但BCUTc-1l（nhigh最低部分电荷加权BCUT）、C1SP2（与另一碳结合的双重追踪碳）、nHBAcc（氢键受体数量（使用CDK HbondAcceptor计数描述符算法））等对生物活性都有负面影响。

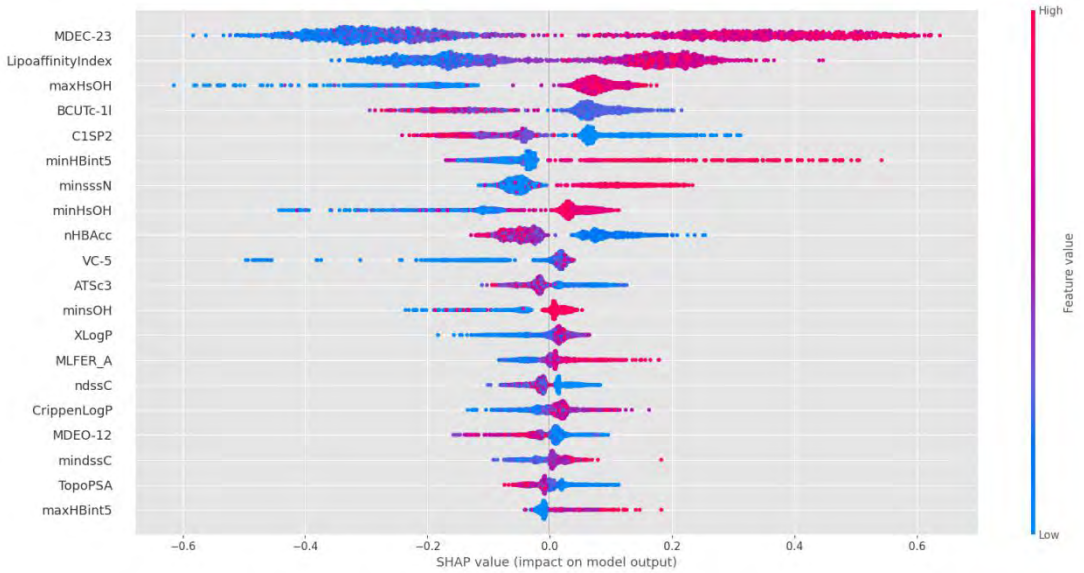


图 4.5 基于随机森林模型的特征重要性排序

4.5 模型小结

针对药物的生物活性影响因素中的变量选择问题，本文首先对变量进行了预处理和相关性分析，通过分析分子描述符之间相关性，发现绝大多数变量不符合正态分布规律，因此采用了 Spearman 相关系数进行相关性分析，初步筛选掉冗余变量和与生物活性相关性较低的变量。随后，采用基于随机森林模型的变量选择算法，结合内置随机森林重要性、基于排列的重要性、SHapley Additive exPlanation 三种特征重要性计算方法进行特征重要性

排序，最终采用 SHapley Additive exPlanation 的特征重要性结果，得到 20 个对生物活性最具有显著影响的分子描述符分别为：MDEC-23、LipoaffinityIndex、maxHsOH、BCUTc-1l、C1SP2、minHBint5、minsssN、minHsOH、nHBAcc、VC-5、ATSc3、minsOH、XLogP、MLFER_A、ndssC、CrippenLogP、MDEO-12、mindssC、TopoPSA、maxHBint5，同时分析了不同描述符对生物活性影响的正负影响程度。

五、问题二：模型的建立与求解

5.1 问题分析

根据问题二的要求得出，该问题的分子描述符变量选择范围为问题一所得到的前 20 个对生物活性最具有显著影响的分子描述符。IC50 值可以由预测得到的 pIC50 转换得到，因此，拟从以下两个步骤解决问题二：（1）构建**决策树、逻辑回归、线性回归、轻度量化梯度提升机（LightGBM）**的 ERα生物活性的定量回归预测模型，为了更好地筛选出重要的描述符，在这一步中选择的变量为问题一中经过相关性筛选后的变量，并且对模型性能进行评估；（2）选择出最优的回归预测模型，比较该模型和随机森林模型得到的前 20 个特征重要性最高的描述符之间的差异性，获得两个排名前 20 的分子描述符集合之间的交叉集合，并且对参数进行调优。最后，基于最终确定的分子描述符和参数构建 ERα生物活性定值回归预测模型，对文件“ERα_activity.xlsx”的 test 表中的 50 个化合物进行 pIC50 值预测，经过对数转换后得到 IC50 值。

5.1 定值回归模型的构建和特征描述符的选择

5.1.1 回归方法调研

针对问题一的要求，需要对 1974 个化合物的 729 个分子描述符进行变量选择，回归分析方法在大数据的挖掘处理过程中起着重要的作用。目前主流的回归分析方法有回归算法、正则化方法、决策树学习、集成算法。对这些回归方法的特点和优缺点进行调研，得到结果如下表 5.1 所示。

表 5.1 回归分析方法比较

	方法描述	优缺点
回归算法	采用对误差的衡量来探索因变量和自变量之间的关系，如线性回归、逻辑回归等	建模迅速，适用于小数据量、简单的关系，对非线性的数据拟合不好
正则化方法	通常是回归算法的延伸，根据算法的复杂度对算法进行调整。如 Least Absolute Shrinkage and Selection Operator（LASSO）、岭回归等	可以防止过拟合和提高模型泛化性能，但会造成欠拟合
决策树学习	根据数据的属性采用树状结构建立决策模型。如分类及回归树（Classification And Regression Tree），C4.5，随机森林（Random Forest）等	具有很高的复杂度和高度的非线性关系，模型容易解释，但存在过拟合倾向，运行速度慢和内存消耗高
集成算法	由多个相对较弱的学习模型独立地就同样的样本进行训练，将预测结果以某种方式整合起来进行总体预测。如 Boosting，Bagging， AdaBoost，梯度推进机（Gradient Boosting Machine, GBM）等	当先最先进的预测几乎都使用了集成算法。精确率高于其他单个模型预测出来的结果

5.1.2 基于 LightGBM 算法的药物优化回归模型的构建

为了得到描述符的特征重要性排序，拟采用集成学习算法—**轻度量化梯度提升机（LightGBM）**构建生物活性值回归模型[4]，同时选择十种经典的机器学习算法作为基线模型，如线性回归模型、LASSO 回归、随机森林、决策树、多层感知机模型等。通过对基线模型进行评估，对比分析出最优的回归模型，进而得到描述符的特征重要性排序。

①基于直方图的决策树算法

LightGBM 是基于 Histogram 的决策树算法，其原理是将连续的浮点型特征离散成 k 个离散值，并构造宽度为 k 的直方图，遍历训练数据后，统计每个离散值在直方图的累计统计量，而在进行特征选择时，只需要根据直方图的离散值，遍历寻找最优的分割点，如图 5.1 所示。

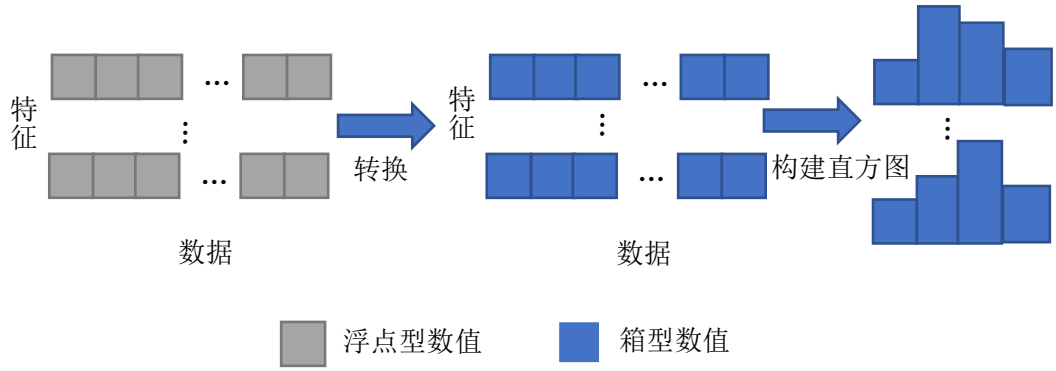


图 5.1 带有深度限制的按叶子生长算法

相比传统的 XGBoost、AdaBoost 自适应提升模型，LightGBM 采用带有深度限制的按叶子生长算法（leaf-wise）代替了按层生成的决策树生长策略，可以降低更多误差，如图 5.2 所示。

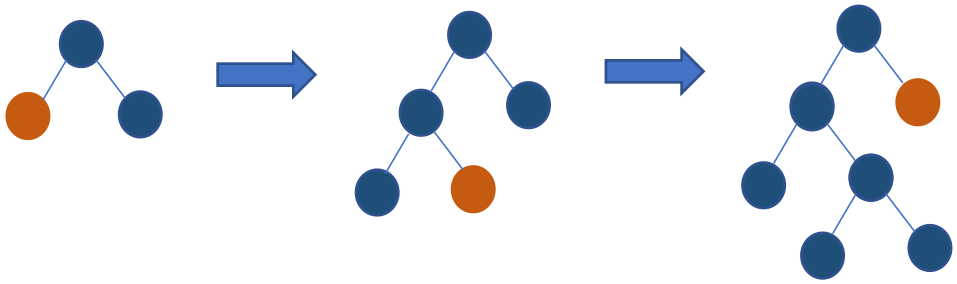


图 5.2 带有深度限制的按叶子生长算法

提升树是利用加模型与前向分布算法实现学习的优化过程，LightGBM 算法能过很好的通过减少特征量而不影响精确度的问题，它主要包含两个算法单边梯度采样（Gradient-based One-Side Sampling）和互斥特征绑定算法（Exclusive Feature Bundling）。

②单边梯度采样

对于n个样本的训练集$\{x_1, x_2, x_3, \ldots, x_n\}$，每个样本$x_i$有 m 维特征，模型的每次梯度迭代，变量的损失函数的负梯度方向为$\{g_1, g_2, g_3, \ldots, g_n\}$，决策树通过最大信息增益点将数据分配到各个节点，O 表示某个固定节点的训练集，分割特征 j 的分割点 p 定义为：

$$V_{j|O}(d) = \frac{1}{n_O}\left(\frac{(\sum_{\{x_i \in O: x_i <= d\}} g_i)^2}{n^j_{l|O}(d)} + \frac{(\sum_{\{x_i \in O: x_i > d\}} g_i)^2}{n^j_{r|O}(d)}\right) \quad (5\text{-}1)$$

$$n_O = \sum I[x_i \in O] \quad (5\text{-}2)$$

$$n^j_{l|O}(d) = \sum I[x_i \in O: x_i \geq d] \quad (5\text{-}3)$$

$$n^{j}_{r|O}(d) = \sum I[x_i \in O: x_i > d] \quad (5\text{-}4)$$

遍历每个特征的分裂节点，找到$d^*_j = \text{argmax}_d V_j(d)$并且计算最大的信息增益值，然后根据数据特征$j^*$的分裂节点将数据分到左右子节点。在 GOSS 中，根据数据的梯度降序排列训练集，保留最高的 a 个数据示例，作为数据子集 A，对于剩下的样本进行随机采样获得子样本 B，最后，信息增益的公式如下：

$$\widetilde{V}_j(d) = \frac{1}{n}\left(\frac{(\sum_{\{x_i \in A: x_{ij} <= d\}} g_i + \frac{1-a}{b}\sum_{x_i \in B: x_{ij} <= d} g_i)^2}{n^j_l(d)} + \frac{(\sum_{\{x_i \in A: x_{ij} > d\}} g_i + \frac{1-a}{b}\sum_{x_i \in B: x_{ij} > d} g_i)^2}{n^j_r(d)}\right) \quad (5\text{-}5)$$

③互斥特征绑定算法

在抗乳腺癌候选药物优化中，药物数据和描述符数据往往具有特征量多并且特征空间系数的特点，互斥特征绑定算法可以通过特征捆绑的方式减少特征维度，对于互斥特征，LightGBM 使用直方图算法对这些特征进行合并。在两个特征并不是完全互斥的情况下，可以用一个指标对特征不互斥的程度进行衡量，从而得到冲突比率，当冲突比率较小时，可在不影响最后精度的前提下，将不完全互斥的特征进行捆绑。

5.1.3 多种回归模型的评估

采用平均绝对误差（MAE）、均方误差（MSE）、均方根误差（RMSE）和 R2（R-Square）决定系数来确定最优回归模型，计算公式分别如 X 所示。

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \quad (5\text{-}6)$$

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (5\text{-}7)$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \quad (5\text{-}8)$$

$$\text{R}^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} \quad (5\text{-}9)$$

12 个回归模型的评估结果如图 5.3 所示， LightGBM 在四个指标上的表现均最好，其次是随机森林模型，ExtraTree 模型则表现最差。

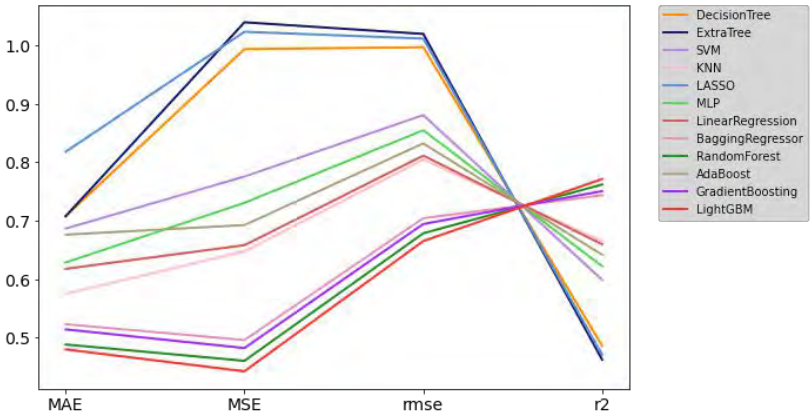


图 5.3 十二类回归模型的评估结果

在 LightGBM 模型下，基于 1974 个样本数据的 196 个变量，绘制如图 5.4 所示的原值和预测值的对比情况。其中纵坐标对应预测结果，横坐标对应真实值，通过真实值和预测值拟合一条直线，可以发现散点和直线大致重合。

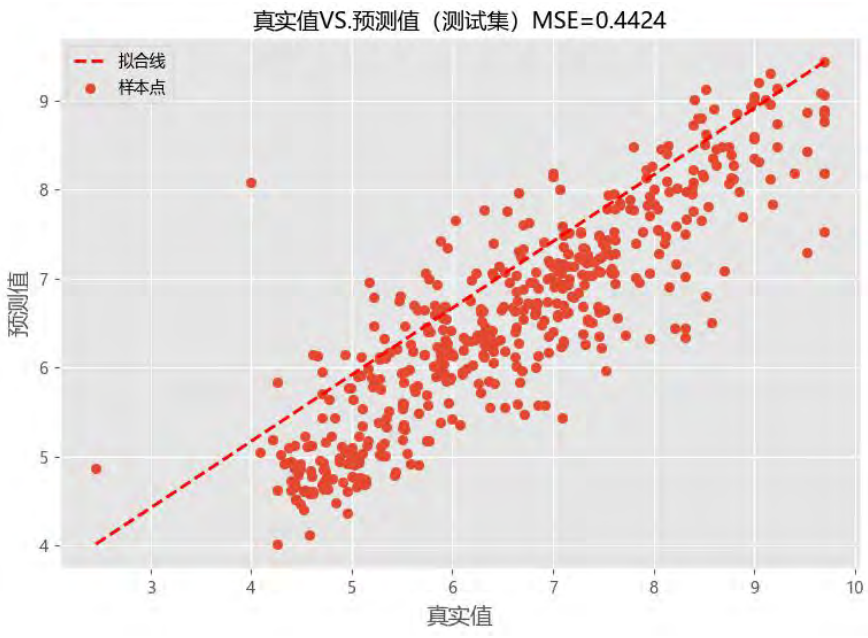


图 5.4 LightGBM 预测值与真实值的拟合图

为了进一步筛选特征进行定值预测，本文对由训练得到的 LightGBM 和 RandomForest 前 20 个重要特征进行排序，如表 5.2 所示，可以发现在两个模型均呈现出重要性的特征有 15 个，由于 LightGBM 算法的预测性能最优，因此本文将选择这 15 个特征构建基于 LightBGM 回归模型进行定值预测。

表 5.2 排名前 20 的特征描述符

分子描述符	SHAP Value（LightGBM）	分子描述符	SHAP Value（RandomForest）
MDEC-23	0.212996386	**MDEC-23**	0.312698075
LipoaffinityIndex	0.19047351	**LipoaffinityIndex**	0.178193013
minsssN	0.119665159	**maxHsOH**	0.116011042
maxHsOH	0.114416838	**BCUTc-1l**	0.09790052
minHsOH	0.087980536	**C1SP2**	0.093620064
MLFER_A	0.083830187	**minHBint5**	0.082440564
BCUTc-1l	0.07847371	**minsssN**	0.073858329
nHBAcc	0.065768763	**minHsOH**	0.070860766
C1SP2	0.061490959	**nHBAcc**	0.066484049
C3SP2	0.050178455	**VC-5**	0.037619015
maxHBint5	0.038272265	ATSc3	0.031815638
AMR	0.036077455	minsOH	0.026887697
XLogP	0.036026191	**XLogP**	0.023957633
minHBint5	0.034860507	**MLFER_A**	0.023897194
MDEO-12	0.032848245	**ndssC**	0.02354074
BCUTp-1h	0.031650292	CrippenLogP	0.023160492
ATSc4	0.030856656	**MDEO-12**	0.019767238
VC-5	0.029655731	mindssC	0.019740989

SaasC	0.028802679	TopoPSA	0.018172734
ndssC	0.02856499	**maxHBint5**	0.01770153

5.2 LightGBM 回归预测模型参数调优

5.2.1 参数调优: max_depth 和 num_leaves

为了进一步提高 LightGBM 回归预测模型的表现，对模型参数的调整十分必要，本文将采用 GridSearchCV（网格搜索）算法对 LightGBM 模型中 learning_rate、num_iterations 等参数进行调整，模型设置的初始参数设置如表 5.3 所示。

表 5.3 LightGBM 回归预测模型的初始参数设置

学习控制参数名	含义	初始参数设置
num_iterations	模型迭代次数	200
learning_rate	模型每次迭代产生的模型的权重	0.1
min_data_in_leaf	叶子节点可能具有的最小记录数	40
bagging_fraction	模型每次迭代时使用的数据比例	0.6
max_depth	树的最大深度，主要是为了防止模型过拟合	7
feature_fraction	模型每次迭代时使用的特征比例	0.8
max_bin	表示将特征存入的 bin 的最大数量	35

由于树结构对模型结果的影响较大，本文先对参数 max_depth（3-10）和 num_leaves（5-100）进行了较大跨度的的排列组合，采用的模型评分参数为 roc_auc，结果如图 5.7 所示，其中，颜色越深代表模型效果越好。从图 5. 5 中可以看出，理想的参数值分别为 6 和 25，随后将模型中对应的参数设置为最优解。

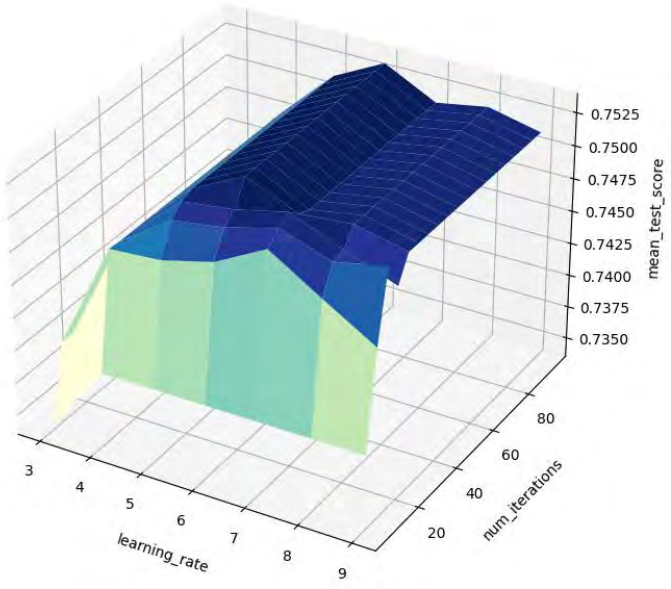


图 5.5 max_depth 和 num_leaves 的调优结果

5.2.2 参数调优: min_data_in_leaf 和 max_bin

继续对 min_data_in_leaf 和 max_bin 的调参，实验结果如图 5.6 所示，从图中可以看出，

理想的参数值分别为 41 和 25，随后将模型中对应的参数设置为最优解。

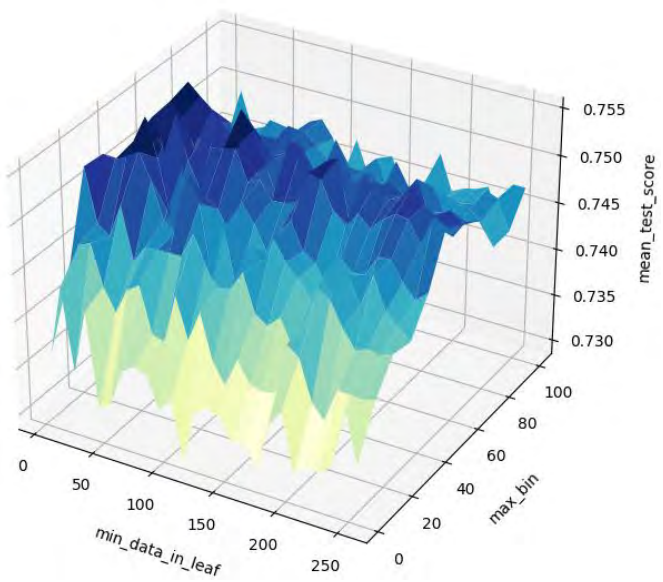


图 5.6 min_data_in_leaf 和 max_bin 的调优结果

5.2.3 参数调优: feature_fraction、bagging_fraction

继续对 feature_fraction 和 bagging_fraction 的调参，实验结果如图 5. 7 所示，从图中可以看出，理想的参数值分别为 0.6 和 0.7，随后将模型中对应的参数设置为最优解。

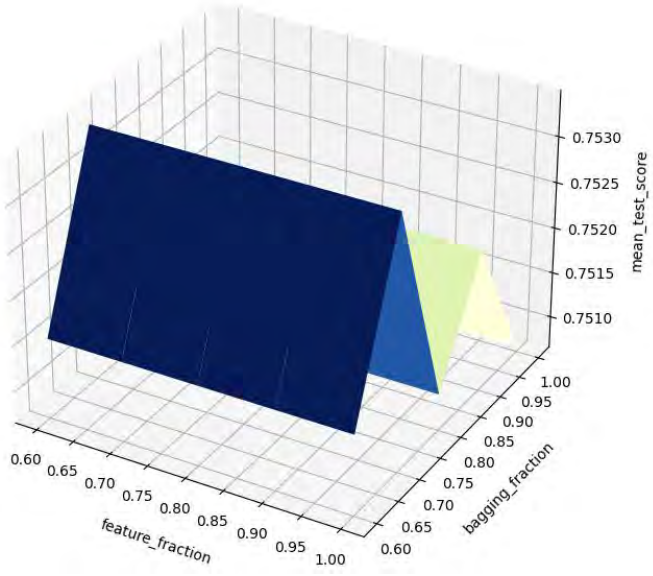


图5.7 feature_fraction、bagging_fraction的调优结果

最后，模型的参数设置如表5.4所示。

表 5.4 LightGBM 回归预测模型的初始参数设置

学习控制参数	调优后的参数值
num_iterations	200
learning_rate	0.1
min_data_in_leaf	41

bagging_fraction	0.6
max_depth	7
feature_fraction	0.7
max_bin	25

5.3 药物活性的回归模型的预测结果分析

基于调参后的 LightGBM 回归预测模型，对药物的生物活性进行定值预测，其部分结果如表 5.5 所示，其他具体结果将以附件的形式提交。

表 5.5 预测值部分结果（按顺序）

序号	IC50_nM	pIC50
1	86.6	7.06257241
2	37.7	7.42406178
3	43.0	7.36633577
4	28.4	7.54670575
5	28.1	7.55059604
6	32.5	7.48817534
7	39.5	7.40335513
8	53.6	7.27076437
9	29.1	7.5356404
10	68.7	7.1628954
11	66.4	7.17798664
12	92.5	7.03362433
13	33.1	7.48023969
14	58.6	7.23203116
15	37.3	7.42832184

5.4 模型小结

在本问题中，本文分别使用了决策树、逻辑回归、线性回归、梯度提升回归算法对经过问题一中相关性筛选后的变量进行建模，并采用 MSE、RMSE、R^2 等指标对十二种回归模型的性能进行评估，评估结果发现，基于 LightGBM 算法的生物活性定值回归预测模型表现效果最好，计算基于 LightGBM 算法的分子描述符的 SHAP 值，并与问题一得到的前 20 个对生物活性最具有显著影响的分子描述符进行对比，选择交叉的 15 个分析描述符作为特征，对模型的参数（如 max_depth 和 num_leaves）进行调整，得到性能最优的生物活性定值回归预测模型，最后，对文件“ERα_activity.xlsx”的 test 表中的 50 个化合物进行 pIC50 值预测，并通过 pIC50 值计算对应的 IC50 值。

六、问题三：模型的建立与求解

6.1 问题分析和技术路线图

针对问题三，需要建立 5 个不同的分类预测模型，并预测文件“ADMET.xlsx”中 50 个化合物对应的不同的 ADMET 值。为解决这个问题，需要进行模型的筛选和训练，模型筛选和训练涉及指标筛选、模型筛选、模型训练、模型评估等过程，因此本文依据模型训练过程确定了该题的研究思路如下：

1. 由于数据集涉及变量过多，为优化模型，使用最优分箱算法计算 729 个变量的**证据权重（Weight of Evidence,WOE）和信息值（Information Value,IV）**，进而确定作为模型输入的重要变量。

2. 寻找在该领域中分类表现比较好的逻辑回归模型（logistics regression,LR）、决策树分类模型（DecisionTree）、随机森林分类模型（RandomForest）、自适应增强分类模型(Adaptive boosting，AdaBoost)、梯度提升分类模型（Gradient boosting，GradientBoosting）、伯努利贝叶斯分类模型（Bernoulli naive Bayes，BernoulliNB）、高斯朴素贝叶斯分类模型（Gaussian naive Bayes，GaussianNB）、支持向量机分类模型（support vector machine,SVM）、K 邻近算法模型（KNeighbors）、神经网络之多层感知器分类模型（Multilayer Perceptron，MLP）、极限梯度提升算法模型（eXtreme Gradient Boosting，XGB）、长短期记忆网络分类模型（Long Short-Term Memory，LSM）、卷积神经网络（Convolutional Neural Network，CNN）等共计 **13 个模型**应用于对 Caco-2、CYP3A4、hERG、HOB、MN 的分类预测，其中有 **11 个机器学习模型、2 个深度学习模型**。

3. 根据模型评估方法，如准确率、召回率、F1 值、混淆矩阵、ROC 曲线、PR 曲线等寻找不同 ADMET 性质的最佳二分类模型。

4. 对最佳二分类模型进行**调参**，以获得不同 ADMET 性质下的最优化模型。

5. 使用**最佳参数的最优二分类模型**，对文件“ADMET.xlsx”的 test 表中的 50 个化合物进行相应的预测。

6. 对模型结果进行进一步可视化分析，包括**特征重要性排序、重要特征变量之间的关系可视化**等，确定不同性质的最优化模型的重要影响变量及变量关系，为问题四的建模提供关于 ADMET 的 5 个性质的基本情况概述。

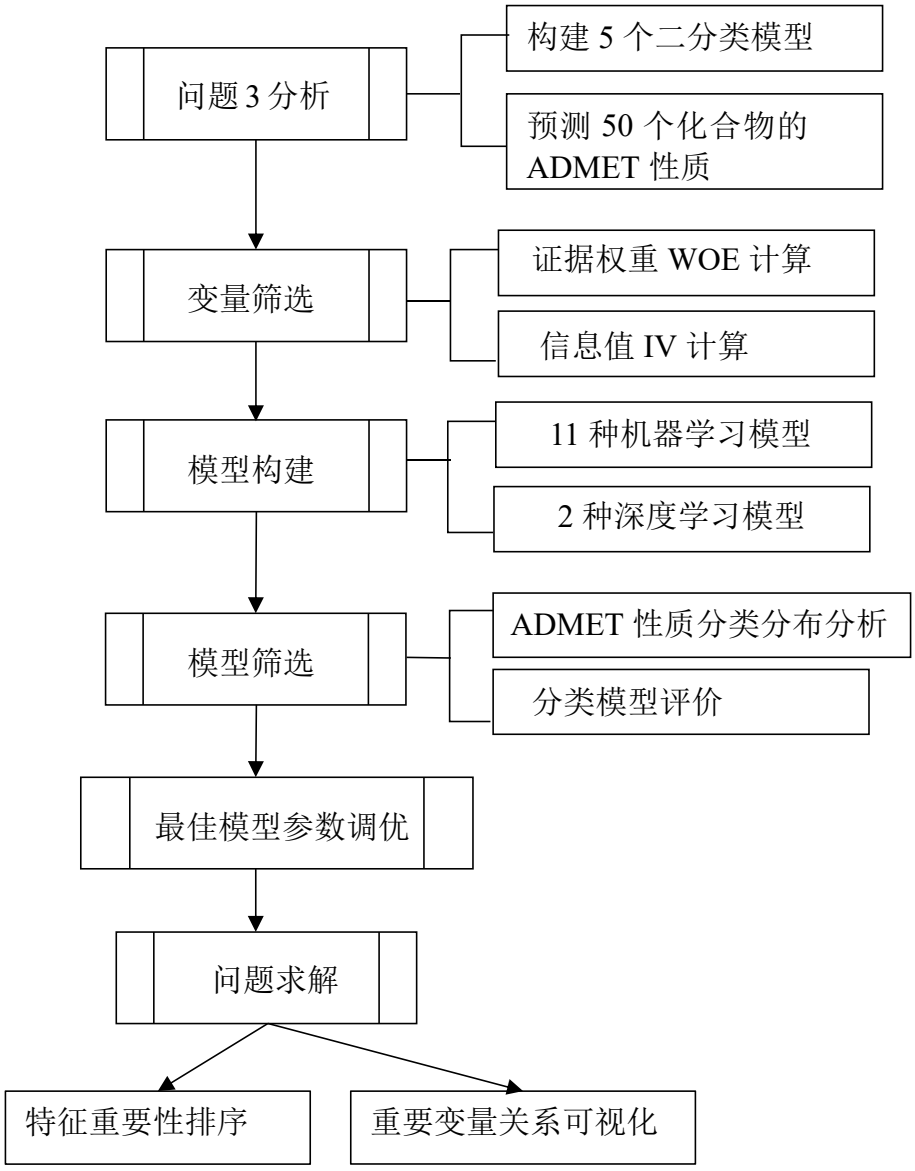


图 6.1 问题三技术路线图

6.2 基于证据权重 WOE 和信息值 IV 的变量筛选

6.2.1 WOE 与 IV 的计算

证据权重（Weight of Evidence)和信息值（Information Value)源于信用评分领域，能够解释自变量相对于因变量的预测能力[5]，避免对变量选择的主观性，WOE 主要用于处理特征变量，IV 与 WOE 密切相关，则常用于进行特征筛选。在进行 WOE 分析前，首先需要进行数据分箱操作，即通过分箱操作将所有变量进行离散化处理，在本文中，主要选取卡方分箱对变量进行处理，通过卡方分箱，连续型变量将会变成一系列取值范围同时被赋予相应的 WOE 取值。本文通过 WOE 和 IV 进行变量筛选的整体计算如下：

（1）卡方分箱

在进行卡方分箱时，首先对变量进行初始化处理，即变量的离散化，然后对箱子进行合并操作，计算公式如（6-1）所示。（6-1）为相邻两个箱子的卡方值，根据卡方独立性检验原理可知，当卡方值越低的时候，则说明两个类别之间的影响程度越低，则可以将这

两个箱子合并；反之，越高，则不能合并。公式中的 m 表示分类变量个数。

$$x^2 = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{(A_{ij}-E_{ij})^2}{E_{ij}} \quad (6\text{-}1)$$

（2）WOE 计算

卡方分箱结束后，计算每个变量取值区间的 WOE 值，WOE 计算公式如（6-2）所示。其中，G_i表示每个分箱中的标签为 0 的数量，G_T则表示标签为 0 的总数量，同理，B_i表示每个分箱中的标签为 1 的数量，B_T则表示标签为 1 的总数量。

$$\mathrm{WOE} = \ln\frac{\mathrm{Count}(B_i)}{\mathrm{Count}(B_T)}\Big/\frac{\mathrm{Count}(G_i)}{\mathrm{Count}(G_T)} \quad (6\text{-}2)$$

（3）IV 计算

IV 为信息值，主要衡量某个变量的信息量，同时表明了某个变量的预测能力，当其值越高的时候，说明该变量对于因变量的影响越大。其计算如（6-3）所示，其中 i 表示第 i 个分箱，n 表示变量分箱个数。

$$\mathrm{IV} = \sum_{i=1}^{n}\left(\frac{\mathrm{Count}(B_i)}{\mathrm{Count}(B_T)} - \frac{\mathrm{Count}(G_i)}{\mathrm{Count}(G_T)}\right) * \mathrm{WOE}_i \quad (6\text{-}3)$$

（4）变量筛选

计算完所有变量信息值之后，需要对变量进行筛选，目前广受认证的筛选标准由 Siddiqi [6]提出，其认为当 IV 小于 0.02 的时候，说明该变量对建模没有任何作用，当 IV>0.2 的时候说明其对建模是有用的。根据该划分标准，确定影响不同 ADMET 性质的重要变量。

6.2.2 变量筛选

根据 6.2.1 的计算方法，本文通过 python 构建相应的模型，以变量对 Caco-2 的影响为例，一共获得 1586 个特征区间，部分特征如表 6.1 所示，“VAR_NAME”表示自变量名称，即分子描述符；“MIN_VALUE”表示在该箱子中该变量的最小值；“MAX_VALUE”表示在该箱子中该变量的最大值；“COUNT”表示在该箱子中的 Caco-2 为 1 和 0 的总数量；“EVENT”表示在该箱子中在该变量影响下的 Caco-2 为 1 的数量；“EVENT_RATE”表示在该箱子中在该变量影响下的 Caco-2 为 1 的比率；“NONEVENT”表示在该箱子中在该变量影响下的 Caco-2 为 0 的数量；“NON_EVENT_RATE”表示在该箱子中在该变量影响下的 Caco-2 为 0 的比率；“DIST_EVENT”为公式（6-2）的$\frac{\mathrm{Count}(B_i)}{\mathrm{Count}(B_T)}$的值：“DIST_NON_EVENT”为公式（6-2）的$\frac{\mathrm{Count}(G_i)}{\mathrm{Count}(G_T)}$的值；“WOE”的计算如公式（6-2）所示；“IV”的计算如公式（6-3）所示。

基于该特征表，对影响 Caco-2 的变量的 IV 值进行排序，最终得到用于模型输入的变量共计 282 个，对其他 ADMET 的性质进行同样的分析，最终得到各个性质中重要变量的数量如表 6.2 所示，在表 6.2 中同时给予了部分变量。

表 6.1 变量区间对 Caco-2 影响的特征汇总部分表

	VAR_NAME	MIN_VALUE	MAX_VALUE	COUNT	EVENT	EVENT_RATE	NONEVENT	NON_EVENT_RATE	DIST_EVENT	DIST_NON_EVENT	WOE	IV
0	nAcid	0	1	1957	759	0.388	1198	0.612	0.999	0.987	0.012	0.027
1	nAcid	2	4	17	1	0.059	16	0.941	0.001	0.013	-2.304	0.027
2	ALogP	-23.105	0.657	658	271	0.412	387	0.588	0.357	0.319	0.112	0.009
3	ALogP	0.657	1.630	658	254	0.386	404	0.614	0.334	0.333	0.004	0.009
4	ALogP	1.633	5.182	658	235	0.357	423	0.643	0.309	0.348	-0.119	0.009

5	ALogp2	0.000	1.560	988	398	0.403	590	0.597	0.524	0.486	0.075	0.006
6	ALogp2	1.569	533.841	986	362	0.367	624	0.633	0.476	0.514	-0.076	0.006
7	AMR	54.067	88.304	498	362	0.727	136	0.273	0.476	0.112	1.447	2.372
8	AMR	88.319	114.821	489	317	0.648	172	0.352	0.417	0.142	1.080	2.372
9	AMR	114.854	141.372	493	68	0.138	425	0.862	0.089	0.350	-1.364	2.372
10	AMR	141.441	517.429	494	13	0.026	481	0.974	0.017	0.396	-3.143	2.372
11	apol	30.662	44.432	495	346	0.699	149	0.301	0.455	0.123	1.311	2.102
12	apol	44.439	59.901	498	328	0.659	170	0.341	0.432	0.140	1.126	2.102
13	apol	59.902	74.421	490	66	0.135	424	0.865	0.087	0.349	-1.392	2.102
14	apol	74.424	359.663	491	20	0.041	471	0.959	0.026	0.388	-2.691	2.102
15	naAromAtom	0	12	655	459	0.701	196	0.299	0.604	0.161	1.319	1.532
16	naAromAtom	13	16	359	191	0.532	168	0.468	0.251	0.138	0.597	1.532
17	naAromAtom	17	18	592	73	0.123	519	0.877	0.096	0.428	-1.493	1.532
18	naAromAtom	19	30	368	37	0.101	331	0.899	0.049	0.273	-1.723	1.532
19	nAromBond	0	15	672	467	0.695	205	0.305	0.614	0.169	1.292	1.034
20	nAromBond	16	18	919	248	0.270	671	0.730	0.326	0.553	-0.527	1.034
21	nAromBond	19	34	383	45	0.117	338	0.883	0.059	0.278	-1.548	1.034

表 6.2 用于预测不同 ADMET 性质的变量个数

ADMET 性质	**变量数量**	**部分主要变量（前 15）**
Caco-2	282	MW,SP-1,ETA_Eta_R_L,WTPT-1,ATSm2,ETA_Beta_s,WPATH,MLFER_L,ETA_Beta,ECCEN,ETA_Alpha，VAdjMat,nHeavyAtom,SP-2,SP-0
CYP3A4	310	Zagreb65,ETA_Eta_R, nHeavyAtom,VAdjMat, ETA_Eta_L,VP-4 ,WTPT-1,VP-2,ETA_Eta_R_L,SP-1,ETA_Beta_s,SP-3,SP-2,ETA_Alpha,VP-3
hERG	290	ETA_Eta_R_L,SP-1 2.267575,CrippenMR,AMR,nBonds,VP-1,VABC,VP-0,apol,McGowan_Volume,fragC,nBondsS,nBonds2,nAtom,bpol
HOB	269	SHBint10,nC,maxHBint10,nHBAcc,fragC, MLogP,BCUTc-1l,hmin,nsOH,nHsOH,MLFER_A,minsOH,SHsOH,maxsOH,SsOH
MN	289	ETA_Psi_1,nHBAcc,XLogP,nN,ETA_Epsilon_2,ETA_Epsilon_5,ETA_EtaP_F,ETA_dEpsilon_C,WTPT-5,TopoPSA,ETA_dEpsilon_A,ETA_Epsilon_4,ETA_Epsi

6.3 模型建立

在本章节中，为了分别构建化合物的 Caco-2、CYP3A4、hERG、HOB、MN 的分类预测模型，本文选择了 11 种流行的机器学习模型和 2 种深度学习模型。

6.3.1 11 种机器学习模型建立

由于机器学习模型种类过多，在进行模型建立叙述过程中仅叙述在后续实验中表现较好的 2 种机器学习模型。本文所选择的 11 种分类模型分别为逻辑回归模型（logistics regression,LR）、决策树分类模型（DecisionTree）、随机森林分类模型（RandomForest）、自适应增强分类模型(Adaptive boosting，AdaBoost)、梯度提升分类模型（Gradient boosting，GradientBoosting）、伯努利贝叶斯分类模型（Bernoulli naive Bayes，BernoulliNB）、高斯朴素贝叶斯分类模型（Gaussian naive Bayes，GaussianNB）、支持向量机分类模型（support vector machine,SVM）、K 邻近算法模型（KNeighbors）、神经网络之多层感知器分类模型（Multilayer Perceptron，MLP）、极限梯度提升算法模型（eXtreme Gradient Boosting，XGB）。

（1）XGBoost 模型

XGBoost(Extreme Gradient Boosting)全名为极限梯度提升，是一种集成学习的机器学习算法，相较于其他集成学习算法，其具有高效灵活的可移植性特点，能够很好地解决目前工业界大规模数据的问题[7]，该算法的核心在于改进了 GBDT 只用一阶导数信息的问题，使得模型求解的效率更高。对于 XGBoost 而言，其目标函数如公式（6-4）所示，其中，L 表示损失函数，由真实值和预测值表示，Ω为抑制模型复杂度的正则项，其表示对全部 K 棵树的复杂度进行求和。

$$Obj = L + \Omega = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{i=1}^{K} \Omega(f_i) \qquad (6\text{-}4)$$

之后，对目标函数进行泰勒二阶展开，求损失函数中的一阶导数和二阶导数的值，再优化目标函数，这也是 XGBoost 最大的特点，就是保留了泰勒展开的二次项，同时为了防止过拟合，XGBoost 使用 Shrikage 方法降低过拟合的概率，再构造树模型的时候也使用随机森李忠的特征子集来确定最优分裂点。

（2）随机森林模型分类模型

同 XGBoost 一样，随机森林分类模型也是集成学习的机器学习算法，其主要思想是使用分类投票机制，将投票最多的类别作为最终的输出类别，是最简单的集成思想，在本文应用随机森林模型，可假设原始数据有 n 个化合物，m 个分子描述符，使用重抽样方法在 n 个化合物中有放回地随机抽取 n_{tree} 个的样本集，同时随机抽取 m_l 个分子描述符，将对分类有影响的分子描述符作为一个分支，进而构建 n_{tree} 个分类树，且每个树都在最大限度增长，不做任何剪枝。在该模型中，n_{tree} 和 m_l 都是很重要的参数。进行树的划分时，其所采用的评估标准是基尼系数，公式(6-5)所示，p_k表示样本为第 K 个类别的概率。

$$Gini(P) = 1 - \sum_{K=1}^{K} (p_k)^2 \qquad (6\text{-}5)$$

6.3.2 LSTM、CNN 两种深度学习模型建立

深度学习模型是近些年人工领域的热点模型，在本文中，尝试选取了短期记忆网络分类模型（Long Short-Term Memory，LSM）、卷积神经网络（Convolutional Neural Network，

CNN）构建模型。

（1）LSTM 模型

LSTM 模型[8]衍生于递归神经网络 RNN 模型，能够学习一种长期的规律，如图 6.2 所示，为 RNN 模型和 LSTM 模型，从图中可以看到同 RNN 相比，LSTM 具有更复杂的结构，在标准的 RNN 模型中仅有 tanh 层，而在 LSTM 中，除了 tanh 层，其还设置了门结构，即 LSTM 可以选择通过信息的节点，包括忘记门、输入门，输出门等，是由西格玛(Sigmoid)神经网络层和逐点乘法运算组成。

使用 LSTM 进行分类的主要计算公式如下，在这些公式中，f 表示忘记门，i 表示输入门，o 表示输出门，c 表示 cell 状态激活向量，σ表示 sigmoid 激活函数，为门的权重，b 表示偏置，tanh 为双曲正切函数，W 是权重矩阵。在本文中，本文尝试使用 LSTM 模型对 ADMET 的 5 种性质进行分类建模。

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \tag{6-6}$$

$$i_t = \sigma(W_{fi} * [h_{t-1}, x_t] + b_i) \tag{6-7}$$

$$\breve{C}_t = tanh(W_C * [h_{t-1}, x_t] + b_C) \tag{6-8}$$

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{6-9}$$

$$h_t = o_t tanh(c_t) \tag{6-10}$$

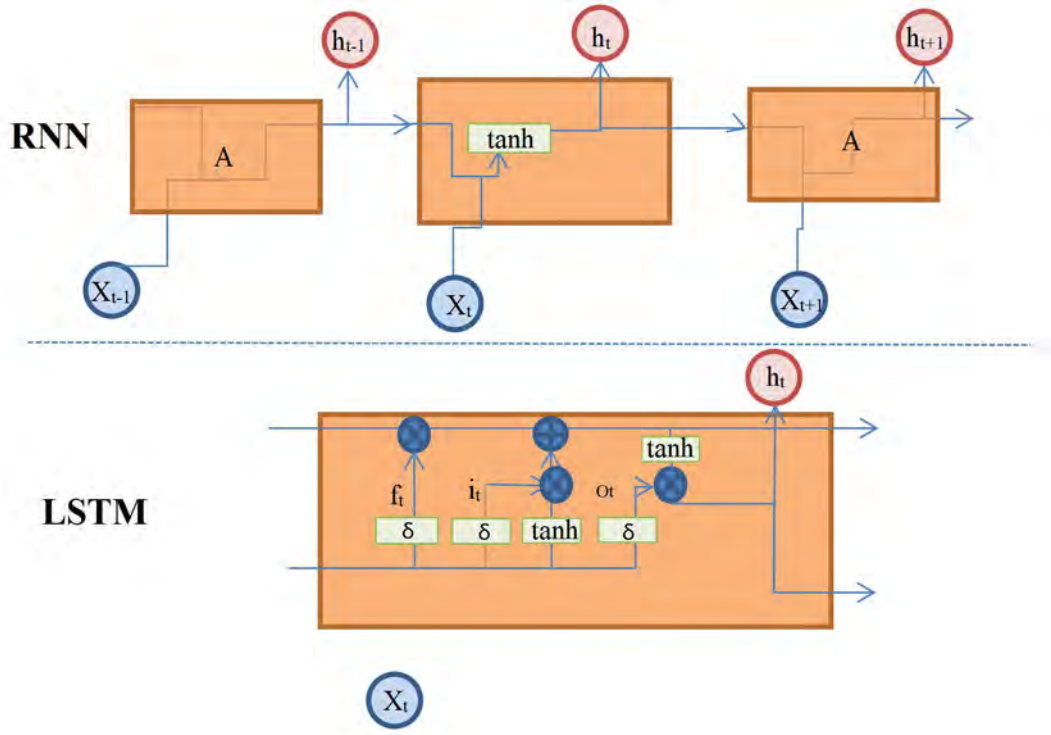


图 6.2 RNN 与 LSTM 模型示意图

（2）CNN 模型

CNN 模型即卷积神经网络(Convolutional Neural Networks，以下简称 CNN),主要是由输入层、卷积层、池化层、全连接层及输出层构成[9]，在本文中，自变量是 729 个分子描述符，需要建立序列模型，因此，使用 CNN 模型进行建模的时候使用的是一维卷积，其模型思路可简单表示为如图 6.3 所示，在一维卷积中，只需要对一维卷积方向中的宽度或者长度的滑动窗口方向进行计算。

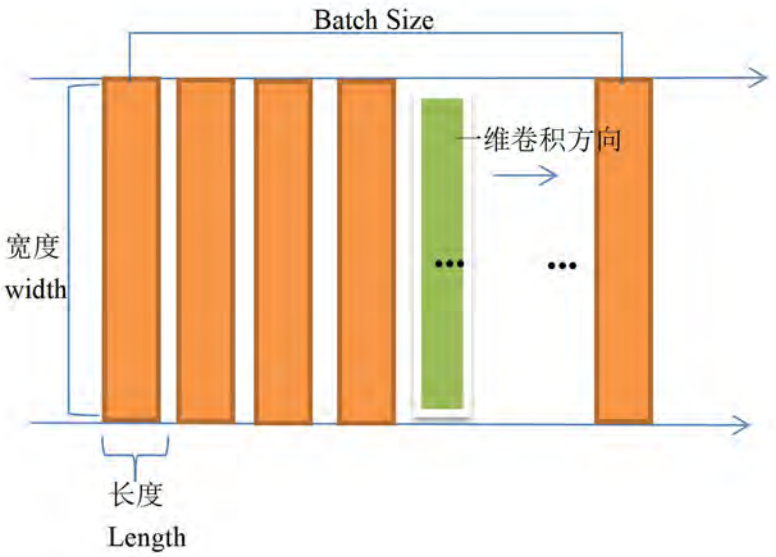


图 6.3 CNN 一维卷积模型示意图

6.4 模型筛选

6.4.1 筛选模型的评价指标

为了筛选出比较好的模型，本文使用以下分类算法评价指标评价各个模型的优劣进行评价。

（1）准确率（Accuracy)

准确率衡量了在所有样本中，预测正确的分类所占有的比例，以本文所使用的数据为例，假设化合物一共 n 种，在模型预测中，被预测为 1 的样本且实际也为 1 的样本为 TP 个，被预测为 1 但实际为 0 的样本为 FP 个，被预测为 0 但实际为 1 的样本为 FN 个，被预测为 0 但实际也为 0 的样本为 TN 个，则准确率的计算如公式（6-11）所示。在模型评估中，其值越大越好。

$$\text{Accuracy} = \frac{TP+TN}{n} \tag{6-11}$$

（2）精度(Precision)

精度是衡量所有样本中，预测为 1 的样本中，预测正确的样本所占有的比例，在模型评估中，其值越大越好。计算如公式（6-12）所示：

$$\text{Precision} = \frac{TP}{TP+FP} \tag{6-12}$$

（3）召回率(Recall)

召回率是衡量在真实样本中，为 1 的样本被预测正确的比率，在模型评估中，其值越大越好。计算公式如（6-13）所示：

$$\text{Recall} = \frac{TP}{TP+FN} \tag{6-13}$$

（4）F 度量（F1）

F1 值是对精度和召回率的加权，被定义为精度和召回率的调和平均数。计算公式如（6-14）所示。在模型评估中，其值越大越好。

$$\text{F1} = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{6-14}$$

（5）PR 曲线

PR 曲线是以精度和召回率分别为横纵坐标的曲线图，在该曲线图中，如果一个模型的 P-R 曲线完全包住另外一个模型的 P-R 曲线，则说明前者模型要优于后者，该曲线图便于可视化。

（6）ROC 曲线和 AUC

ROC 曲线（receiver operating characteristic curve）用来揭示模型敏感性和特异性的相互关系，其 x 坐标和 y 坐标计算如公式（6-15）和（6-16）所示。依据其预测概率和真实分类，可以将不同的(x,y)连接在一起成为 ROC 曲线，而曲线下的面积被称为 AUC（Area under curve），在模型评估中 AUC 越大越好。

$$x = \frac{FP}{FP+TN} \tag{6-15}$$

$$y = \frac{TP}{TP+FN} \tag{6-16}$$

（7）混淆矩阵

混淆矩阵则是由 TP、FP、FN、TN 构成，从混淆矩阵中可以清晰地看到错误和正确的分类情况。

（8）对数损失（LogLoss）

对数损失衡量了模型预测概率和真实概率分布的差异，在模型评估中，其取值越小越好，由于本文构建的是二分类模型，故而其计算公式如（6-17）所示。p_i表示第 i 个样本被预测为 1 的概率。

$$\text{logloss} = -\frac{1}{n}\sum_{i=1}^{n}(y * logp_i + (1-y) * \log(1-p_i)) \tag{6-17}$$

6.4.2 ADMET 性质数据预处理-过采样及样本划分

在进行模型筛选前，需要对数据进行预处理，在进行预处理前，本文将“Molecular_Descriptor.xlsx” train 表格同“ADMET.xlsx”train 表格进行连接，然后观察 ADMET 性质的分类情况，如图 6.4 所示，为“ADMET.xlsx”train 表格中不同性质 0、1 分类情况，从图中可以看到不论是哪一种 ADMET 性质，其数据集的分类是极度不均衡的，因此，需要对样本进行数据平衡处理，在本文中采取的数据均衡处理方法为 **SMOTE 过采样方法**，即通过学习对样本量较少的分类分布规律生成更多与该分类相关的标签样本，进而使 0 和 1 的样本趋于平衡。

然后本文将连接的 train 表格进行进一步的划分，按 8：2 的比例划分为训练集和验证集，并在验证集上进行评估。

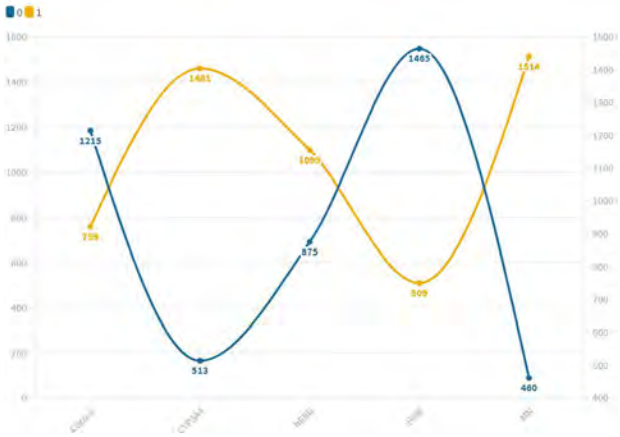


图 6.4 不同 ADMET 性质分类情况

6.4.3 筛选构建 Caco-2 的分类预测模型

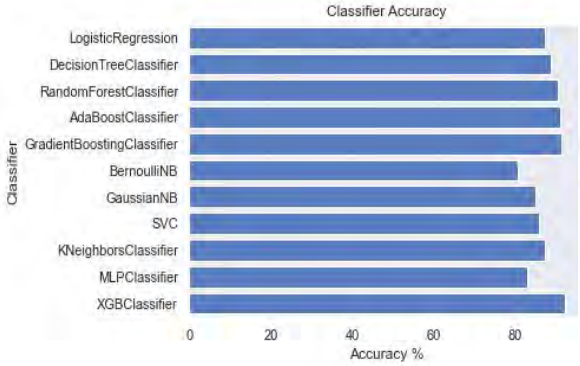
根据 6.2 的变量筛选方法、6.3 所构建的模型及 6.4.2 的数据预处理方法，将预测 Caco-2 的 282 个分子描述符作为自变量输入到 11 种机器学习模型和 2 种深度学习模型中。

如图 6.5 所示为 11 种机器学习模型 Accuracy、LogLoss、ROC 曲线+AUC、P-R 曲线情况，可见在 Caco-2 的分类预测上，大多数模型表现良好，但是如决策树分类模型、高斯朴素贝叶斯模型、伯努利贝叶斯模型、KNN 模型、MLP 模型等其虽然在准确率上较高但是却有很高的 LogLoss，说明预测与真实值之间存在较大的差异，在所有模型中，根据 ROC 曲线和 P-R 曲线，也可以看出相对比较好的分类模型包括 XGBooost、GradientBoost、Adaboost、随机森林分类模型、逻辑回归分类模型等在预测 Caco-2 上具有比较好的表现

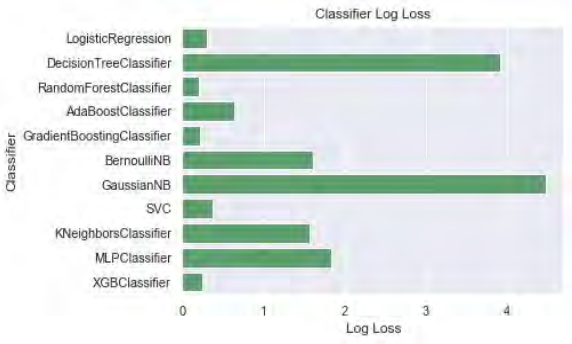
图 6.6 为 11 种机器学习混淆矩阵情况，从矩阵图中可以看到在验证集上，对于 Caco-2 分类情况较好的模型由逻辑回归模型、决策树模型、随机森林模型、AdaBoost、GradientBoost、XGBoost 等，这些模型在真实样本上预测值较好。

如表 6.3 所示，为 11 个机器学习模型的 Precision、Recall、F1-score、Accuracy、Log_loss 的具体数值，图 6.7 为 2 种深度学习模型的表现可视化，我们发现深度学习模型在该数据样本上的表现并不太好，其参数学习率为 0.00001，batchsize 为 64，训练次数为 100 次，后续其他模型的构建设置一样。在 100 次训练后，训练集和测试集（验证集）上的表现差异波动较大，且准确率远远不如机器学习模型，因而在本文中不再对其做进一步的实验和数据记录。

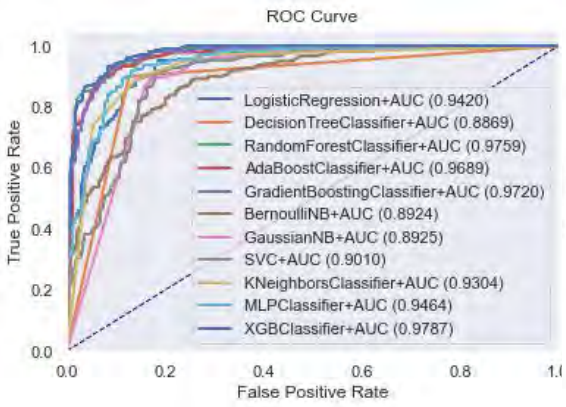
根据所有模型的表现情况，针对 Caco-2 的二分类预测模型，**本文选择 XGBoost 作为其分类模型**，在该模型下，验证集准确率达到了 92%，对数损失为 0.24，两个分类上的召回率、精度和 F1 值都达到了 90%以上，AUC 值为 0.9787。



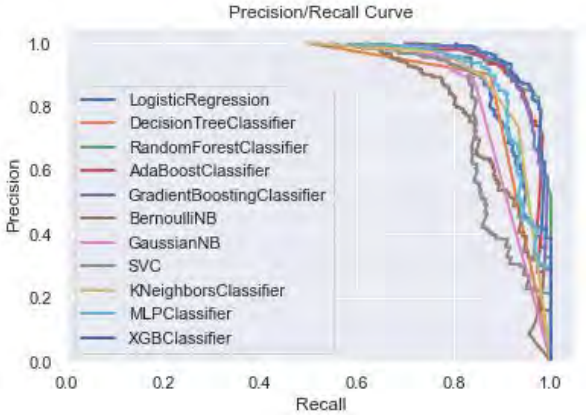
（a）Accuracy



（b）LogLoss



（c）ROC 曲线+AUC



（d）P-R 曲线

图 6.5 11 种机器学习模型 Accuracy、LogLoss、ROC 曲线、P-R 曲线情况

表 6.3 11 种机器学习在验证集上的表现

模型	分类	Precision	Recall	F1-score	Accuracy	Log_loss
LogisticRegression	0	0.89	0.85	0.87	0.87	0.3
	1	0.86	0.9	0.88		
DecisionTree	0	0.9	0.87	0.89	0.89	3.91
	1	0.88	0.9	0.89		
RandomForest	0	0.91	0.91	0.91	0.91	0.21
	1	0.9	0.9	0.90		
AdaBoost	0	0.91	0.92	0.91	0.91	0.64
	1	0.92	0.9	0.91		
GradientBoosting	0	0.92	0.91	0.91	0.91	0.22
	1	0.91	0.92	0.91		
BernoulliNB	0	0.79	0.84	0.81	0.81	1.61
	1	0.83	0.77	0.8		
GaussianNB	0	0.89	0.8	0.84	0.85	4.48
	1	0.82	0.9	0.86		
SVC	0	0.91	0.8	0.85	0.89	0.38
	1	0.82	0.92	0.87		
KNeighbors	0	0.89	0.86	0.87	0.87	1.57
	1	0.86	0.89	0.88		
MLPClassifier	0	0.98	0.68	0.8	0.82	1.83
	1	0.75	0.98	0.85		
XGBClassifier	**0**	**0.93**	**0.91**	**0.92**	**0.92**	**0.24**
	1	**0.91**	**0.93**	**0.92**		

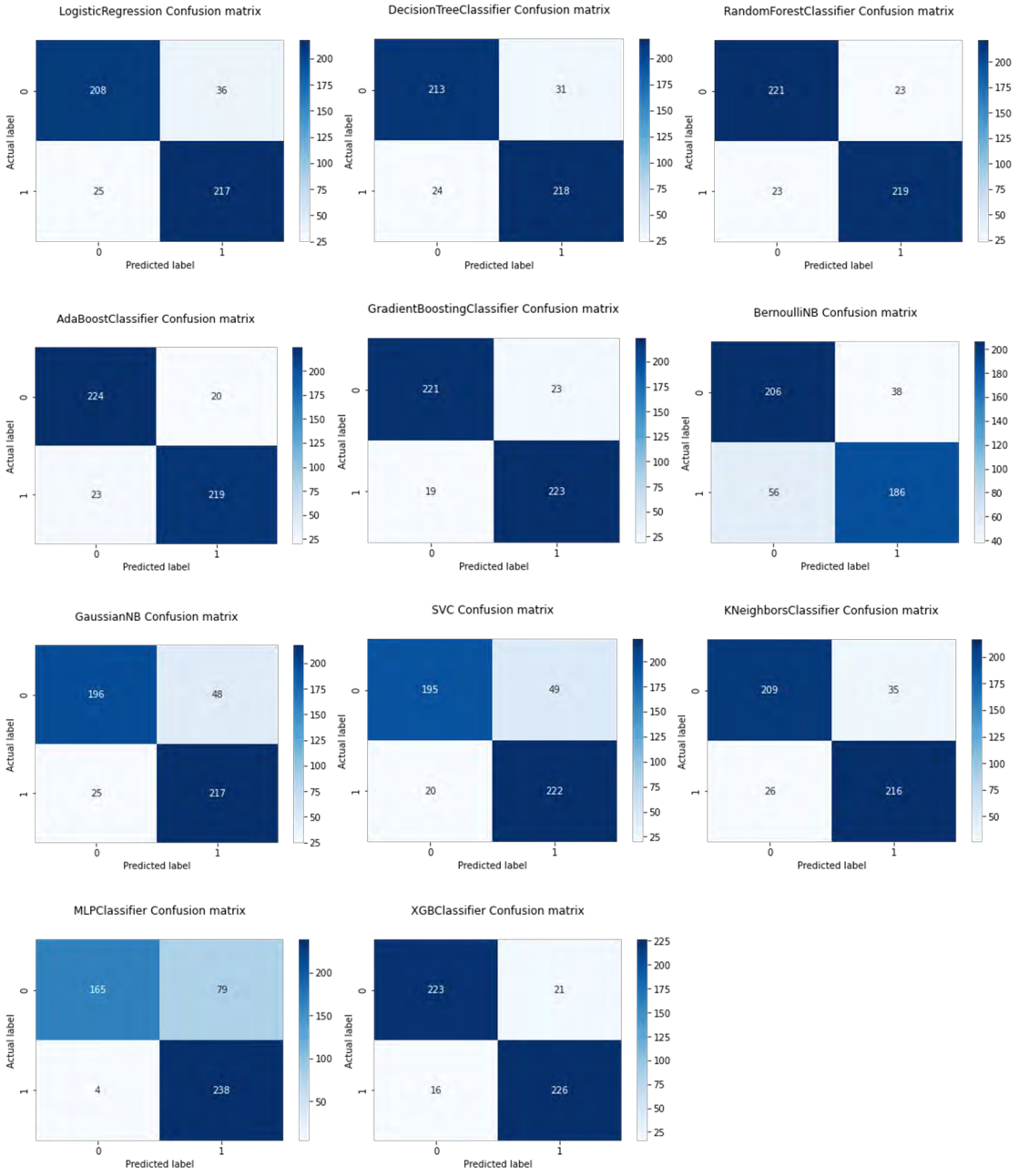
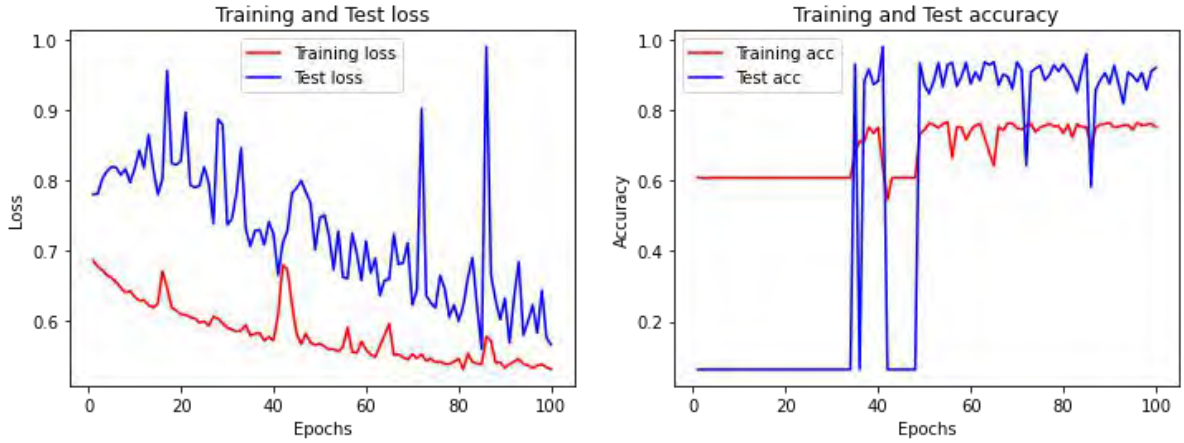
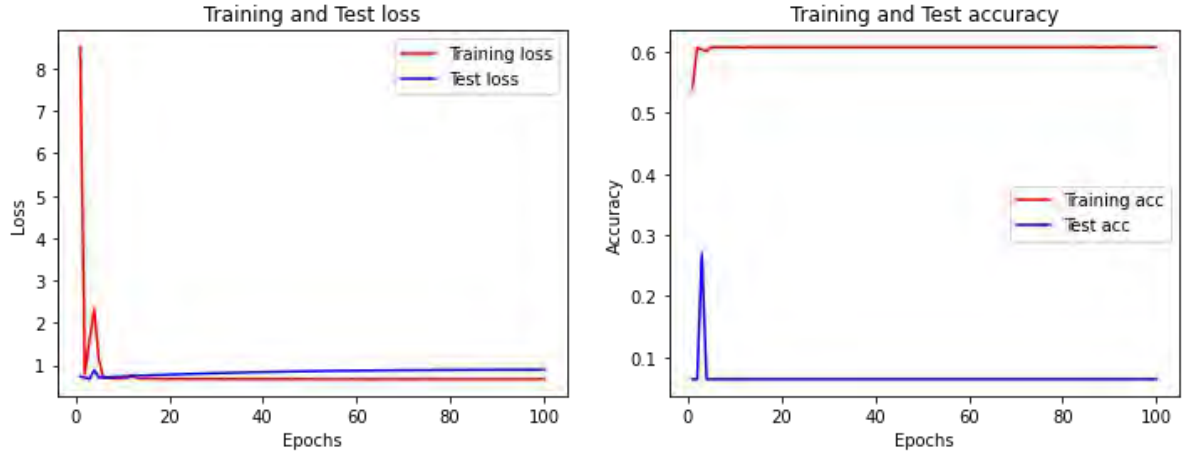


图 6.6 11 种机器学习模型混淆矩阵情况



（a） LSTM 的 Accuracy 和 Log_Loss



(b) CNN 的 Accuracy 和 Log Loss

图 6.7 2 种深度学习模型 Accuray 和 logloss 情况

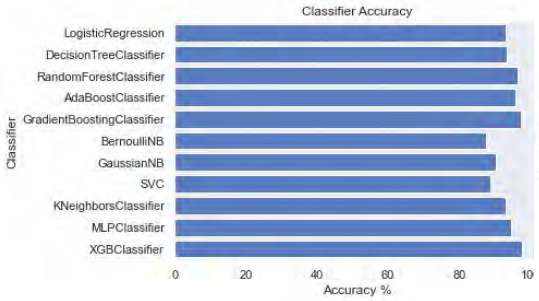
6.4.4 筛选构建 CYP3A4 的分类预测模型

在筛选构建 CYP3A4 的分类预测模型模型上，所使用的方法和实证同 6.4.3 筛选构建 Caco-2 的分类预测模型是一致的。表 6.4 为机器学习模型表现情况，图 6.8 为其表现情况的可视化，图 6.9 为 2 种深度学习模型的表现情况，针对两种深度学习模型，我们可以显然看到训练集和测试集不仅准确率低，而且 logloss 也很大，且 CNN 的模型表现远不如 LSTM，但无论如何深度学习的模型基本上远不如训练出的最好的机器学习模型，从这些表和图中可以看到，**用于构建 CYP3A4 的分类预测模型最好为 XGBoost 模型。**在该模型下，其在验证集 0 和 1 的分类上都表现出了最优，即准确率到达 0.98，AUC 值为 0.9944，接近于 1。

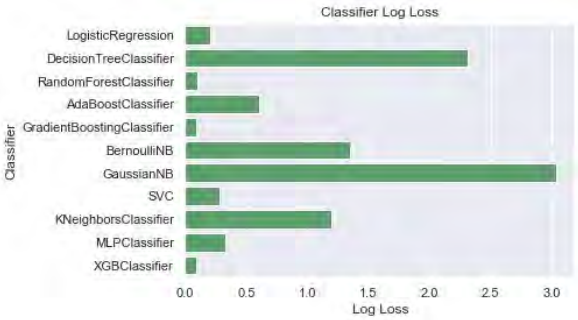
表 6.4 11 种机器学习在验证集上的表现

模型	分类	Precision	Recall	F1-score	Accuracy	Log_loss
LogisticRegression	0	0.9	9.97	0.93	0.93	0.21
	1	0.97	0.89	0.93		
DecisionTree	0	0.93	0.93	0.93	0.93	2.31
	1	0.93	0.94	0.93		
RandomForest	0	0.95	0.98	0.96	0.96	0.1
	1	0.98	0.95	0.96		

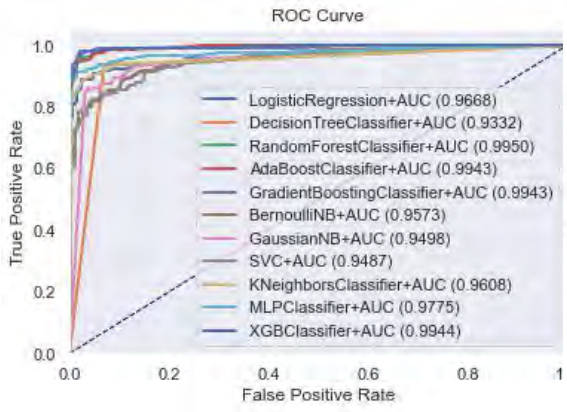
AdaBoost	0	0.96	0.95	0.96	0.958	0.6
	1	0.96	0.96	0.96		
GradientBoosting	0	0.97	0.98	0.97	0.974	0.09
	1	0.98	0.97	0.97		
BernoulliNB	0	0.91	0.83	0.87	0.88	1.35
	1	0.85	0.92	0.88		
GaussianNB	0	0.87	0.95	0.91	0.9	3.04
	1	0.94	0.86	0.9		
SVC	0	0.85	0.94	0.89	0.89	0.28
	1	0.93	0.84	0.88		
KNeighbors	0	0.89	0.97	0.93	0.929	1.19
	1	0.97	0.89	0.93		
MLPClassifier	0	0.91	0.99	0.95	0.947	0.33
	1	0.99	0.91	0.95		
XGBClassifier	**0**	**0.98**	**0.98**	**0.98**	**0.977**	**0.089**
	1	**0.98**	**0.98**	**0.98**		



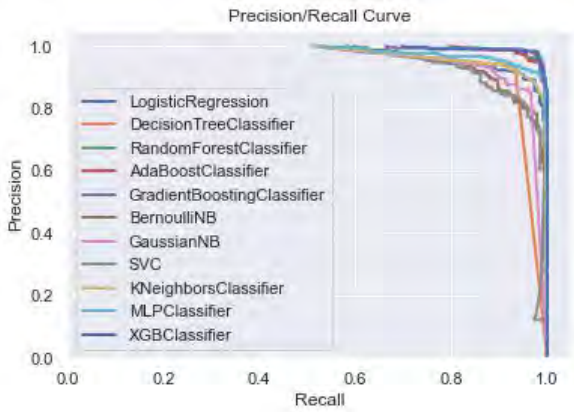
（a）Accuracy



（b）LogLoss

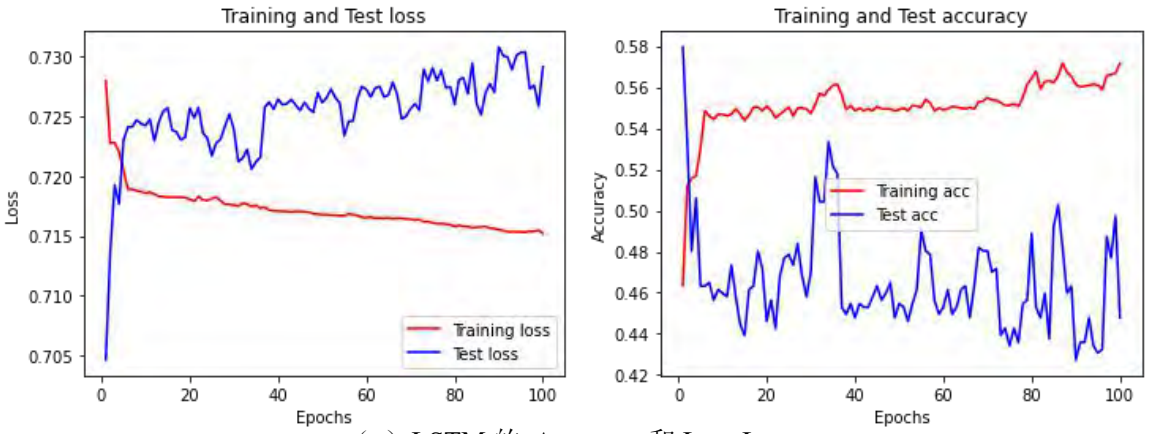


（c）ROC 曲线+AUC

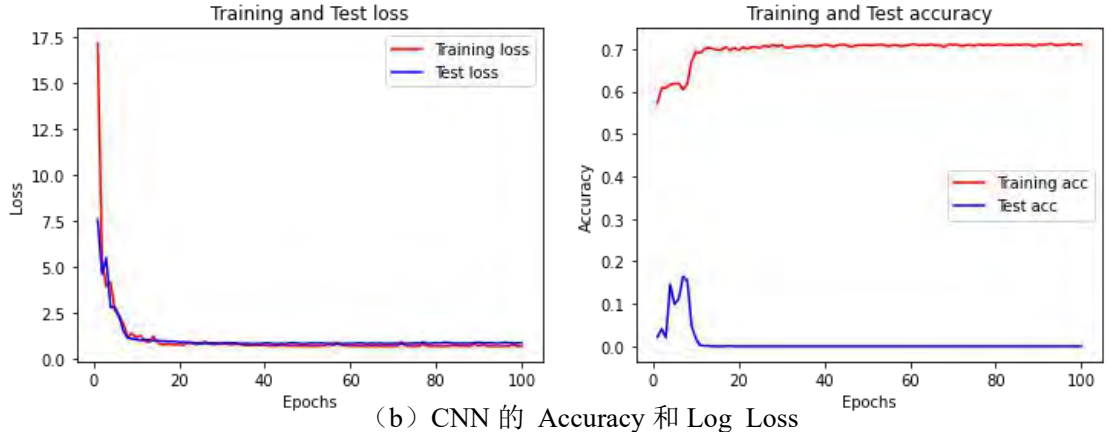


（d）P-R 曲线

图 6.8　11 种机器学习模型 Accuracy、LogLoss、ROC 曲线、P-R 曲线情况



（a）LSTM 的 Accuracy 和 Log_Loss



（b）CNN 的 Accuracy 和 Log Loss

图 6.9 2 种深度学习模型 Accuracy、LogLoss 情况

6.4.5 筛选构建 hERG 的分类预测模型

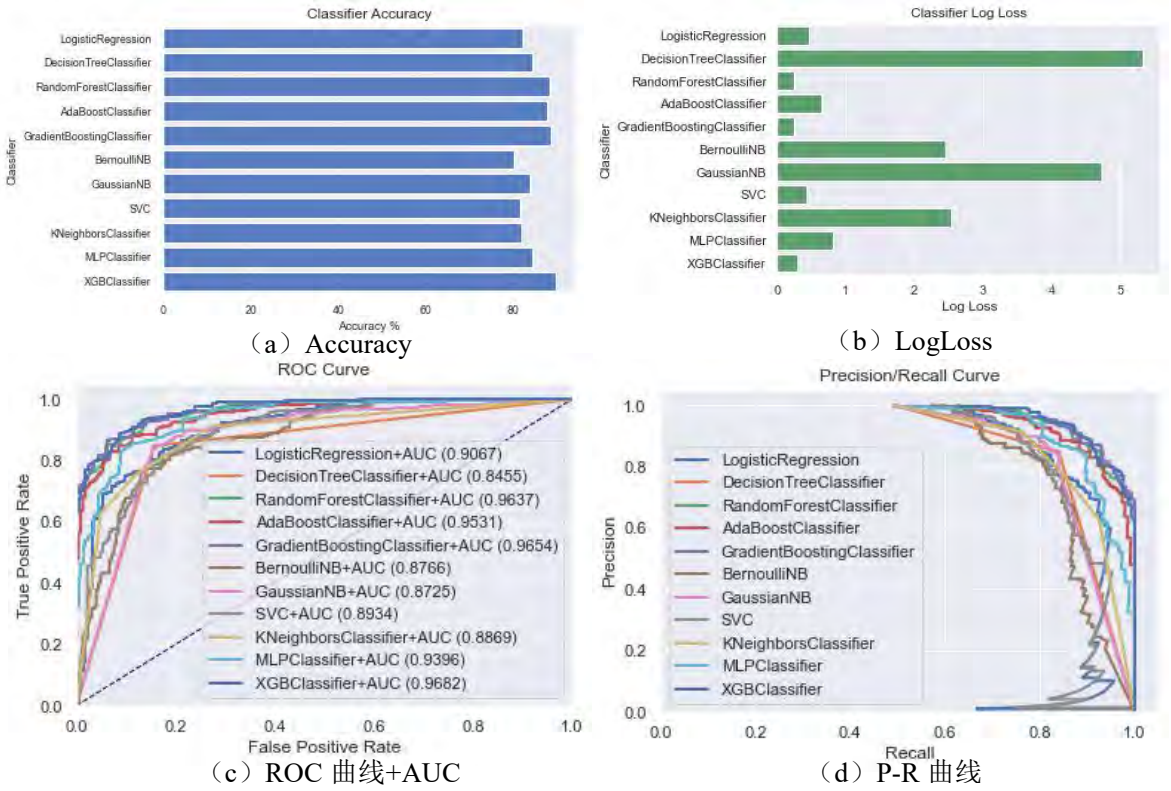
在筛选构建 hERG 的分类预测模型模型上，同理我们计算了 precision、recall、f1-score、accuracy 等值，绘制了 ROC 曲线、PR 曲线、accuracy 曲线、logloss 曲线等。表 6.5 为构建 hERG 分类机器学习模型表现情况，图 6.10 为其表现情况的可视化，图 6.11 为 2 种深度学习模型的表现情况，深度学习模型的表现仍然比较差，但相较于 CYP3A4 的分类模型构建，其表现要相对于好一些，该深度模型预测效果也好于部分机器学习模型，但是其 log_loss 比较大。

综合分析各类图表，可以看出，在构建 hERG 的分类预测模型上，**XGBoost 分类模型**仍然为最好，其 accuracy 为 0.89，log_loss 为 0.288，AUC 为 0.9682。事实上，在建立 hERG 的分类预测模型上，所有的机器学习模型表现不尽人意，均没有达到 90%以上，XGBoost 是在默认参属下相对而言最好的模型，需要进一步进行优化。

表 6.5 11 种机器学习在验证集上的表现

模型	分类	Precision	Recall	F1-score	Accuracy	Log_loss
LogisticRegression	0	0.81	0.84	0.83	0.823	0.465
	1	0.83	0.8	0.82		
DecisionTree	0	0.85	0.84	0.85	0.845	5.339
	1	0.84	0.85	0.84		
RandomForest	0	0.89	0.88	0.89	0.88	0.25
	1	0.88	0.88	0.89		
AdaBoost	0	0.87	0.89	0.88	0.877	0.644

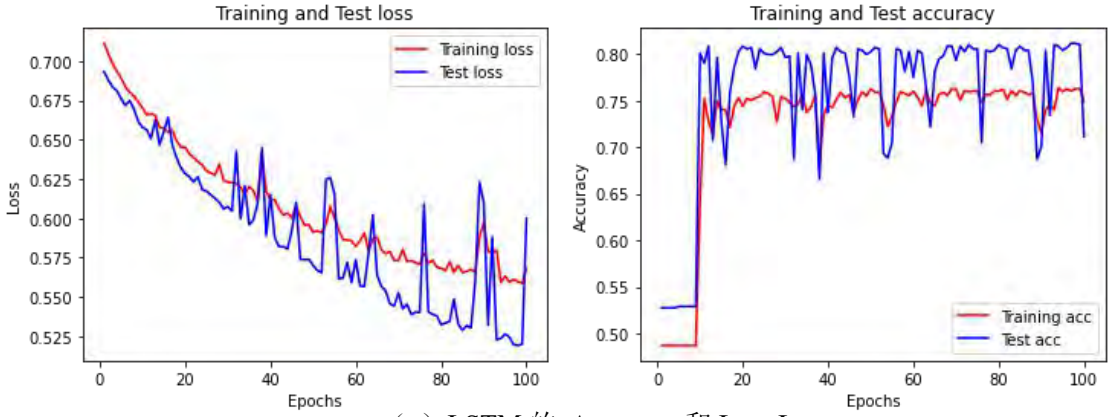
	1	0.88	0.87	0.87		
GradientBoosting	0	0.89	0.89	0.89	0.886	0.242
	1	0.88	0.88	0.88		
BernoulliNB	0	0.83	0.77	0.8	0.8	2.45
	1	0.78	0.83	0.83		
GaussianNB	0	0.84	0.84	0.84	0.838	4.737
	1	0.83	0.84	0.84		
SVC	0	0.83	0.81	0.82	0.816	0.423
	1	0.81	0.82	0.82		
KNeighbors	0	0.79	0.88	0.83	0.818	2.537
	1	0.86	0.76	0.8		
MLPClassifier	0	0.89	0.78	0.84	0.84	0.82
	1	0.8	0.9	0.85		
XGBClassifier	**0**	**0.9**	**0.9**	**0.9**	**0.89**	**0.288**
	1	**0.9**	**0.89**	**0.9**		



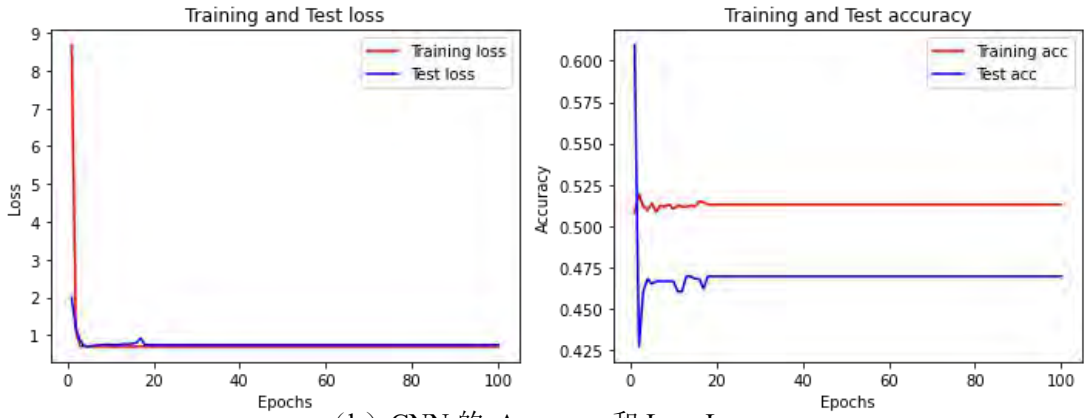
（a）Accuracy　　（b）LogLoss

（c）ROC 曲线+AUC　　（d）P-R 曲线

图 6.10　11 种机器学习模型 Accuracy、LogLoss、ROC 曲线、P-R 曲线情况



（a）LSTM 的 Accuracy 和 Log_Loss



（b）CNN 的 Accuracy 和 Log Loss

图 6.11 2 种深度学习模型 Accuracy、LogLoss 情况

6.4.6 筛选构建 HOB 的分类预测模型

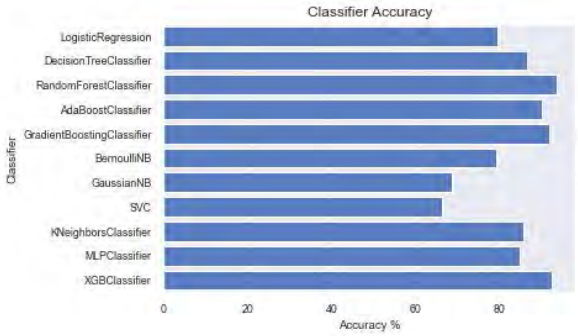
在筛选构建 HOB 的分类预测模型模型上，如前所叙述一致，我们对结果进行了可视化展现，表 6.6 为构建 HOB 分类机器学习模型表现情况，图 6.12 为其表现情况的可视化，图 6.13 为 2 种深度学习模型的表现情况，LSTM 的表现较为有趣，其在训练集上准确率远远低于 0.5，在测试集上的准确率却在 0.9-1 之间变动，模型波动较大，稳定性不好。因此，在构建 HOB 的分类预测模型上，并不考虑深度学习模型。

综合分析各类图表，可以看出，在构建 HOB 的分类预测模型上， RandomForest 分类模型相较于前面 XGBoost 模型是最好的，**因此，在构建 HOB 的二分类预测模型上，选择随机森林分类模型**，其 accuracy 为 0.937，log_loss 为 0.2，AUC 为 0.978。

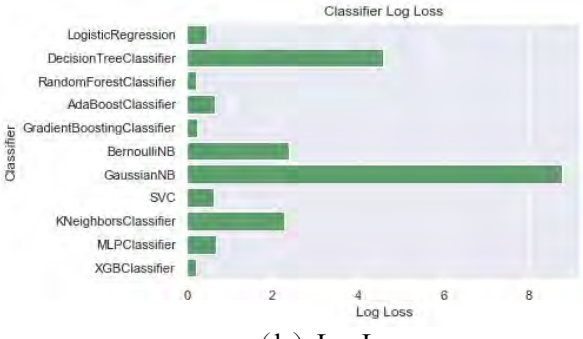
表 6.6 11 种机器学习在验证集上的表现

模型	分类	Precision	Recall	F1-score	Accuracy	Log_loss
LogisticRegression	0	0.79	0.8	0.79	0.795	0.458
	1	0.8	0.79	0.8		
DecisionTree	0	0.85	0.88	0.87	0.867	4.597
	1	0.88	0.85	0.87		
RandomForest	**0**	**0.94**	**0.93**	**0.94**	**0.937**	**0.2**
	1	**0.93**	**0.94**	**0,94**		
AdaBoost	0	0.89	0.91	0.9	0.901	0.656
	1	0.91	0.89	0.9		
GradientBoosting	0	0.91	0.92	0.92	0.918	0.24

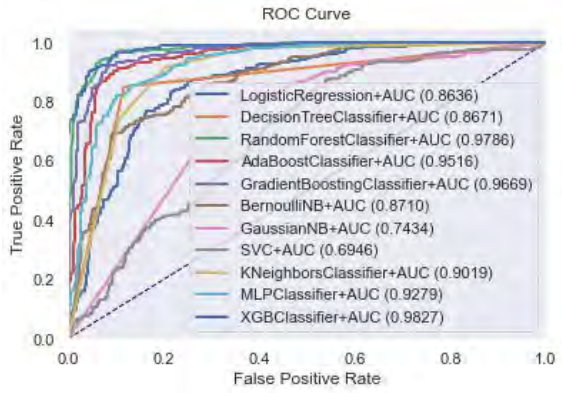
	1	0.92	0.92	0.92		
BernoulliNB	0	0.75	0.87	0.81	0.794	2.37
	1	0.85	0.72	0.78		
GaussianNB	0	0.8	0.49	0.61	0.687	8.787
	1	0.64	0.88	0.74		
SVC	0	0.76	0.47	0.58	0.662	0.627
	1	0.62	0.85	0.72		
KNeighbors	0	0.9	0.8	0.85	0.857	2.28
	1	0.82	0.92	0.87		
MLPClassifier	0	0.93	0.75	0.83	0.848	0.675
	1	0.79	0.95	0.86		
XGBClassifier	0	0.92	0.93	0.92	0.924	0.197
	1	0.93	0.92	0.93		



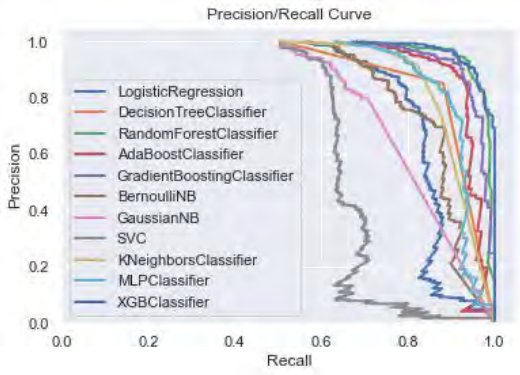
（a）Accuracy



（b）LogLoss

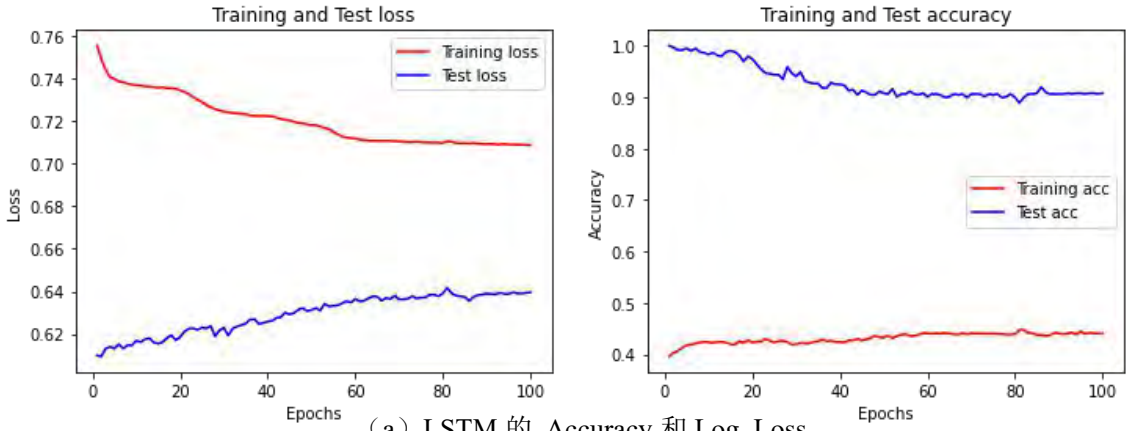


（c）ROC 曲线+AUC

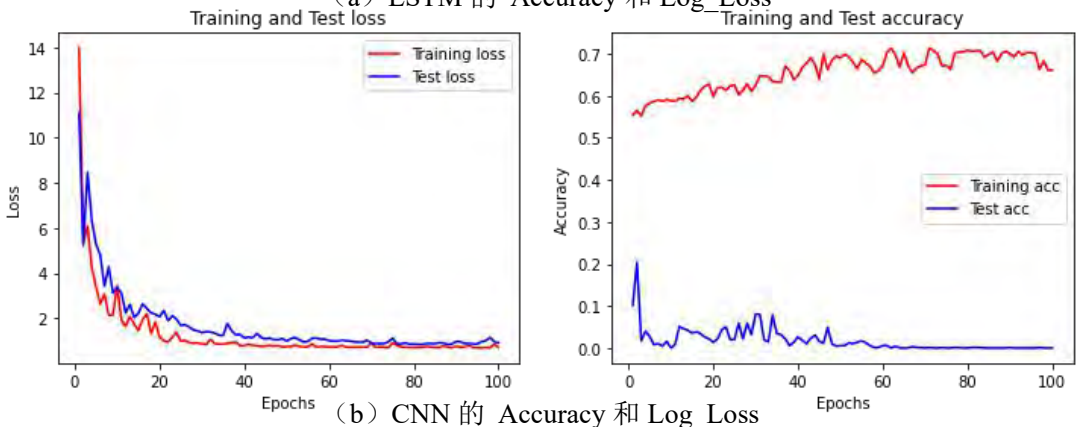


（d）P-R 曲线

图 6.12　11 种机器学习模型 Accuracy、LogLoss、ROC 曲线、P-R 曲线情况



（a）LSTM 的 Accuracy 和 Log_Loss



（b）CNN 的 Accuracy 和 Log_Loss

图 6.13 2 种深度学习模型 Accuracy、LogLoss 情况

6.4.7 筛选构建 MN 的分类预测模型

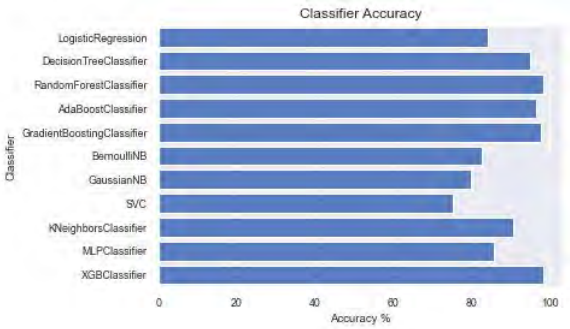
在筛选构建 MN 的分类预测模型模型上，如前所叙述一致，我们对结果进行了可视化展现，表 6.7 为构建 MN 分类机器学习模型表现情况，图 6.12 为其表现情况的可视化，图 6.13 为 2 种深度学习模型的表现情况， LSTM 模型较之 CNN 模型更为稳定，但是其准确率并不高，尤其是在大多数机器学习模型准确率都能到达 0.8 以上的对比下，深度学习模型并不具有优势。

综合分析各类图表，可以看出，在构建 MN 的分类预测模型上， RandomForest 分类模型和 XGBoost 分类模型都比较好，RandomForest 准确率 0.983，loss 为 0.1，AUC 为 0.9954；XGBoost 准确率为 0.981，loss 为 0.07，AUC 为 0.9969。整体上而言在构建 MN 的分类预测模型上，RandomForest 分类模型和 XGBoost 分类模型差别并不大，考虑到前面已构建了 3 个 XGBoost 分类模型，**在 MN 的分类预测上，本文选择使用随机森林分类预测模型。**

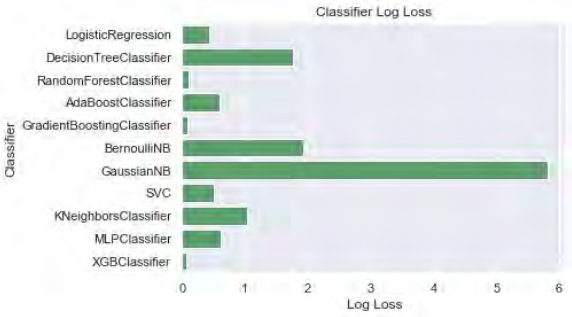
表 6.7 11 种机器学习在验证集上的表现

模型	分类	Precision	Recall	F1-score	Accuracy	Log_loss
LogisticRegression	0	0.82	0.86	0.84	0.842	0.434
	1	0.86	0.82	0.84		
DecisionTree	0	0.94	0.96	0.95	0.949	1.767
	1	0.96	0.94	0.95		
RandomForest	**0**	**0.98**	**0.99**	**0.98**	**0.983**	**0.1**
	1	**0.99**	**0.98**	**0.98**		
AdaBoost	0	0.95	0.98	0.96	0.963	0.594

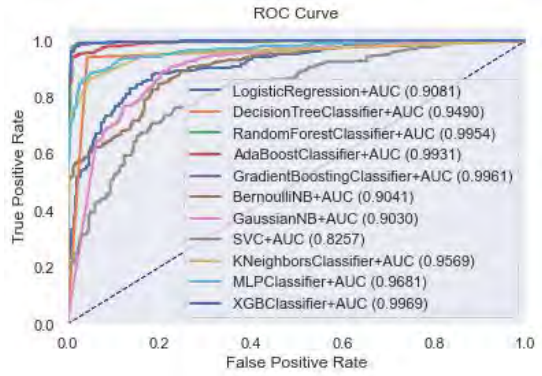
	1	0.98	0.95	0.96		
GradientBoosting	0	0.97	0.99	0.98	0.978	0.086
	1	0.99	0.97	0.98		
BernoulliNB	0	0.84	0.8	0.82	0.825	1.931
	1	0.82	0.85	0.83		
GaussianNB	0	0.75	0.9	0.81	0.798	5.825
	1	0.88	0.71	0.78		
SVC	0	0.75	0.73	0.74	0.751	0.51
	1	0.75	0.77	0.76		
KNeighbors	0	0.87	0.95	0.91	0.906	1.04
	1	0.95	0.86	0.9		
MLPClassifier	0	0.95	0.75	0.84	0.856	0.611
	1	0.8	0.96	0.87		
XGBClassifier	**0**	**0.98**	**0.98**	**0.98**	**0.981**	**0.07**
	1	**0.98**	**0.98**	**0.98**		



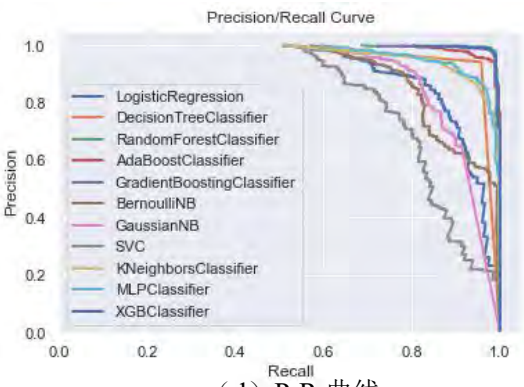
（a）Accuracy



（b）LogLoss

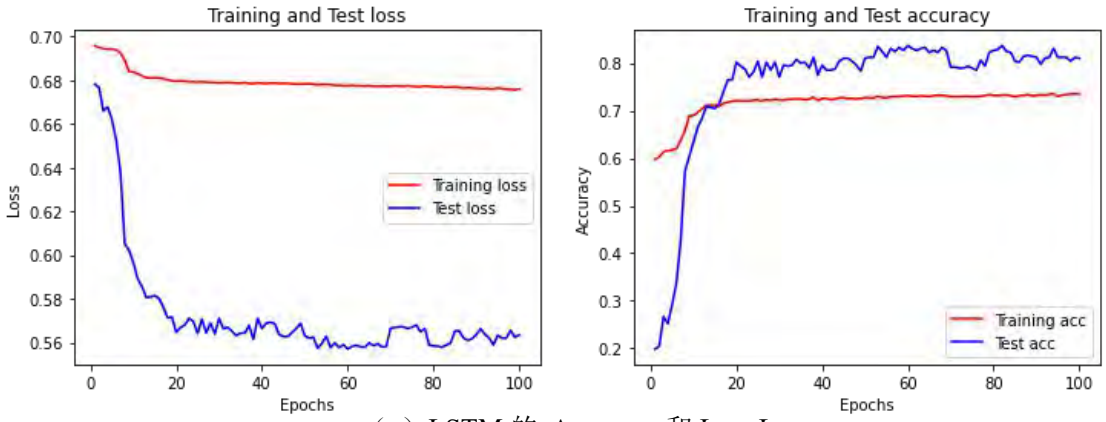


（c）ROC 曲线+AUC

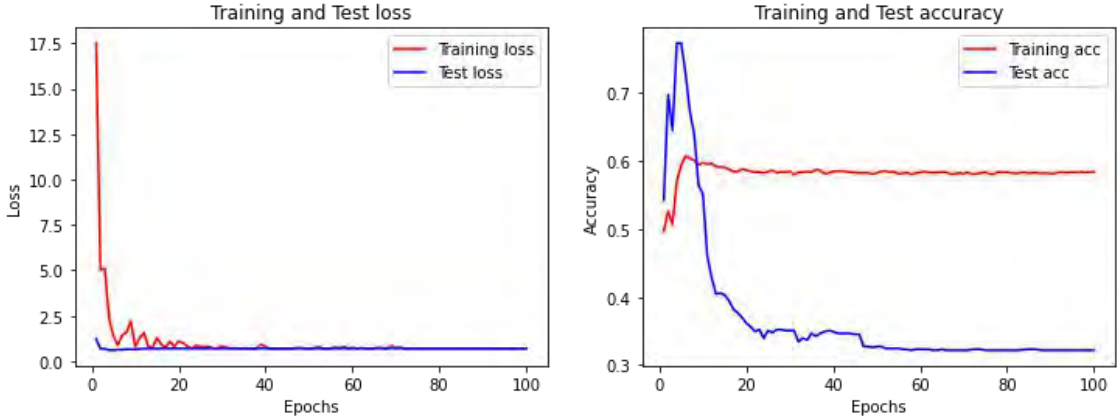


（d）P-R 曲线

图 6.14　11 种机器学习模型 Accuracy、LogLoss、ROC 曲线、P-R 曲线情况



（a）LSTM 的 Accuracy 和 Log_Loss



（b）CNN 的 Accuracy 和 Log Loss

图 6.15 2 种深度学习模型 Accuracy、LogLoss 情况

6.5 基于 XGBoost 和随机森林分类模型的不同 ADMET 性质模型求解

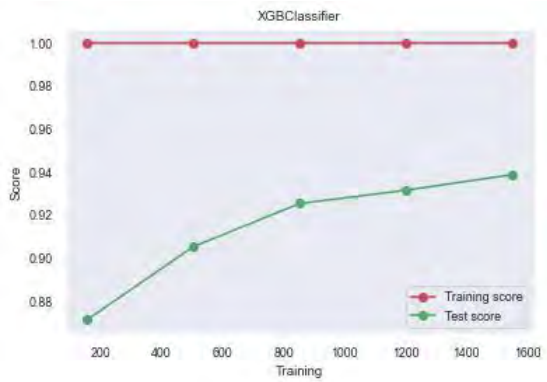
6.5.1 模型参数调优

根据 6.4 章节模型筛选，我们确定了 Caco-2、CYP3A4、hERG、HOB、MN 的分类预测模型的最佳二分类模型，如表 6.8 所示。

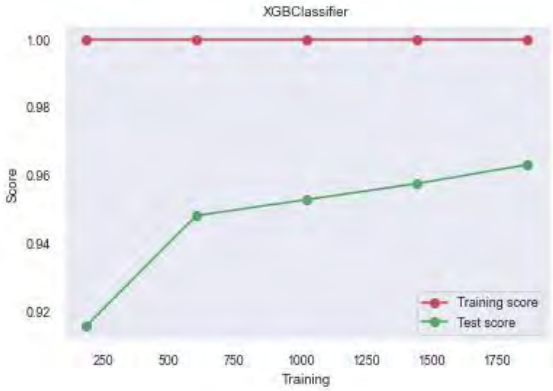
表 6.8 不同 ADMET 性质最佳二分类模型

ADMET 性质	最佳二分类模型
Caco-2	XGBoost
CYP3A4	XGBoost
hERG	XGBoost
HOB	RandomForest
MN	RandomForest

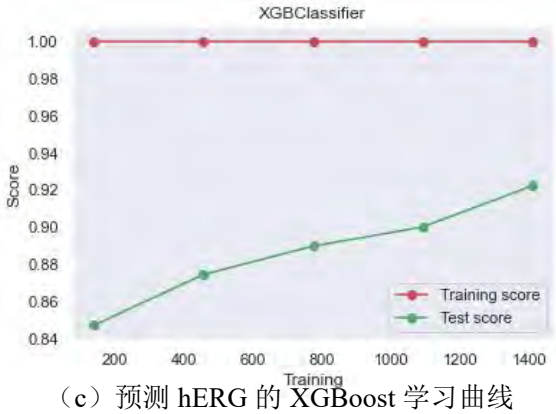
为了观察各个最佳模型的泛化能力，本文绘制了各个最佳模型的学习曲线，如图 6.16 所示，从图中可以看出，最佳模型在训练集上的表现十分优秀，展现出了各个模型超强的学习能力，在测试集上，针对不同 ADMET 性质的分类预测模型表现出了不同的泛化能力，总体而言，各个模型在测试集的表现都较好，学习分数都超过了 0.9，其中，预测 MN 的 RanomForest 分类模型和预测 CYP3A4 的 XGBoost 分类模型学习分数基本上超过 0.96，表现出了极强的模型泛化能力。



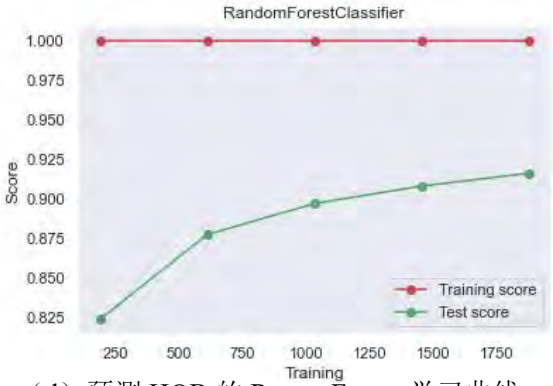
（a）预测 Caco-2 的 XGBoost 学习曲线



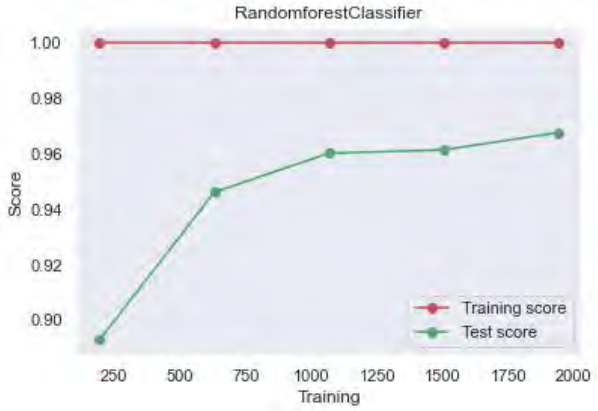
（b）预测 CYP3A4 的 XGBoost 学习曲线



（c）预测 hERG 的 XGBoost 学习曲线



（d）预测 HOB 的 RanomForest 学习曲线



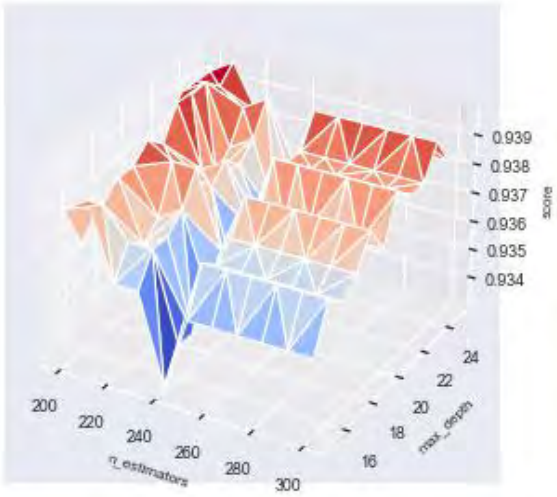
（e）预测 MN 的 RanomForest 学习曲线

图 6.16 不同 ADMET 性质分类预测最佳模型的学习曲线

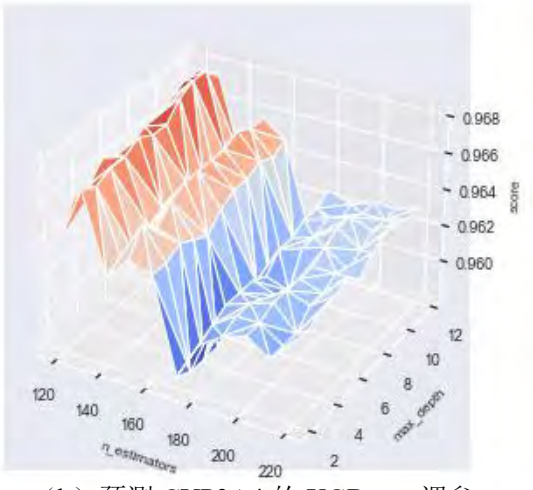
接下来，本文对这些最佳模型进行参数调优，在 6.4 章节中，我们将题目中提供的训练集进一步划分为了训练集和验证集，并在验证集上大致测定了模型的效果，为了进一步进行最佳模型的参数调优，在本文中，本文直接使用训练集进行参数选择，并且使用**十折交叉验证方法**来检验模型的效果。十折交叉验证是将数据集划分了 10 个小数据集，并轮流将其中 9 份数据用于做训练，另外 1 个数据用来做验证，最后，10 次效果的均值作为最终的模型精度，该方法精度高且更能够准确地评估模型的好坏。

如图 6.17 所示，为各个最佳模型的最优参数选择，从图中可以总结出各个最佳模型的最优参数如表 6.9 所示，将基于该表进行模型求解。其中，对于预测 Caco-2 的 XGBoost 调参，预测准确率最高可达到 93.9%；对于预测 CYP3A4 的 XGBoost 调参，预测准确率最高可达到 96.8%；对于预测 hERG 的 XGBoost 调参，预测准确率最高可达到 92.6%；对于

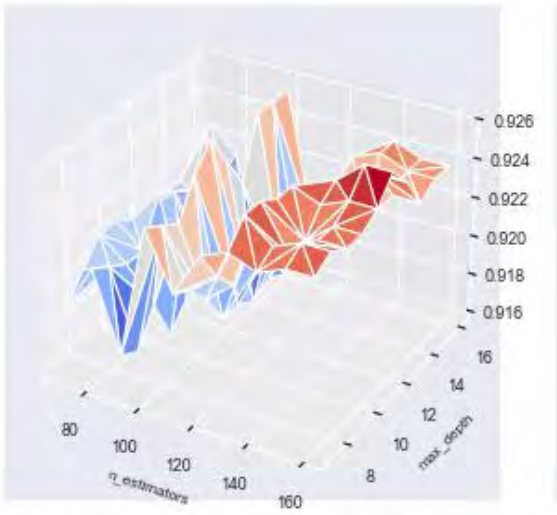
预测 HOB 的 RandomForest 调参，预测准确率最高可达到 92%；对于预测 MN 的 RandomForest 调参，预测准确率最高可达到 97.8%。



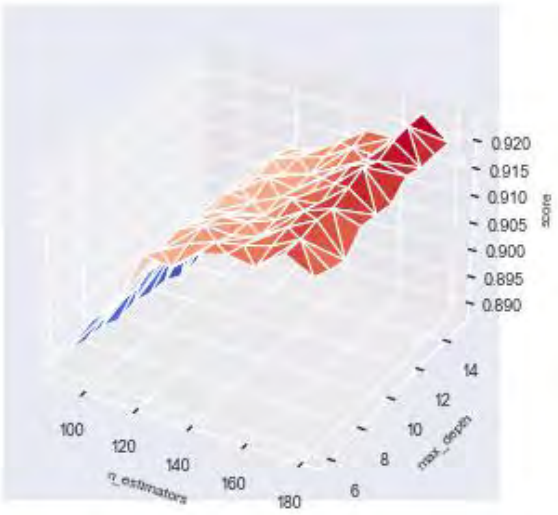
（a）预测 Caco-2 的 XGBoost 调参



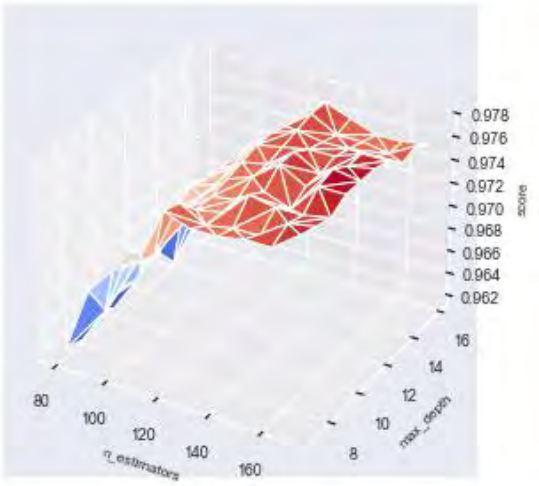
（b）预测 CYP3A4 的 XGBoost 调参



（c）预测 hERG 的 XGBoost 调参



（d）预测 HOB 的 RandomForest 调参



（e）预测 MN 的 RandomForest 调参

图 6.17 最佳模型参数调整

表 6.9 不同 ADMET 性质最佳模型的最优参数

ADMET 性质	最佳二分类模型	最优参数	
		n_estimators	max_depth
Caco-2	XGBoost	200	23
CYP3A4	XGBoost	120	10
hERG	XGBoost	160	13
HOB	RandomForest	180	14
MN	RandomForest	160	12

6.5.2 基于最优 XGBoost 和随机森林分类模型的 ADMET 性质求解

根据题目要求，需要对文件“ADMET.xlsx”的 test 表中的 50 个化合物进行相应的预测，结合“Molecular_Descriptor.xlsx”所提供的分子描述符及前述针对不同性质所筛选出的变量（分子描述符），将针对不同性质所筛选的 50 个化合物的分子描述符输入到经过 6.5.1 模型参数调优后的最佳二分类模型中，最终得到如表 6.10 所示的结果，具体结果以附件形式提交。

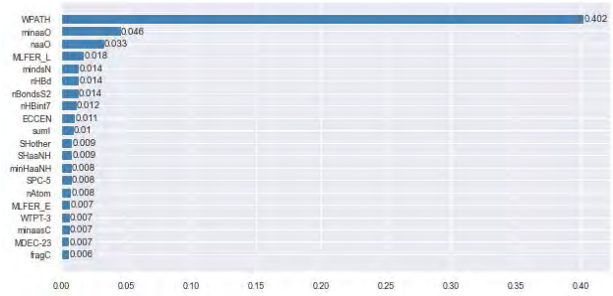
表 6.10　50 个化合物的 ADMET 预测结果数据（按文档里的顺序）

序号	Caco-2	CYP3A4	hERG	HOB	MN	序号	Caco-2	CYP3A4	hERG	HOB	MN
1	0	1	1	0	1	26	1	1	1	1	0
2	0	1	0	0	1	27	0	1	1	0	0
3	0	1	1	0	1	28	0	1	1	0	1
4	0	1	1	0	1	29	0	1	1	0	1
5	0	1	1	0	1	30	0	1	1	1	1
6	0	1	1	0	1	31	1	1	1	1	1
7	0	1	1	0	1	32	1	1	1	1	1
8	0	1	1	0	1	33	1	1	1	1	1
9	0	1	1	0	1	34	1	1	1	1	1
10	0	1	1	0	1	35	0	1	1	1	1
11	0	1	1	0	1	36	0	1	1	0	1
12	0	1	1	0	1	37	0	1	0	0	1
13	0	1	1	0	1	38	0	1	1	0	0
14	0	1	1	0	1	39	0	1	0	0	1
15	0	1	1	0	1	40	0	1	0	0	1
16	0	1	1	0	1	41	0	1	0	0	1
17	0	1	1	0	1	42	0	1	0	0	1
18	0	1	0	0	1	43	0	1	0	0	1
19	0	1	0	0	1	44	0	1	0	0	1
20	0	0	0	0	1	45	0	1	0	0	1
21	0	1	1	0	1	46	0	1	1	0	1
22	0	1	0	0	1	47	0	1	1	0	1
23	1	0	1	0	0	48	0	1	1	0	1
24	1	0	1	0	0	49	0	1	1	0	1
25	1	1	1	1	0	50	0	1	1	0	0

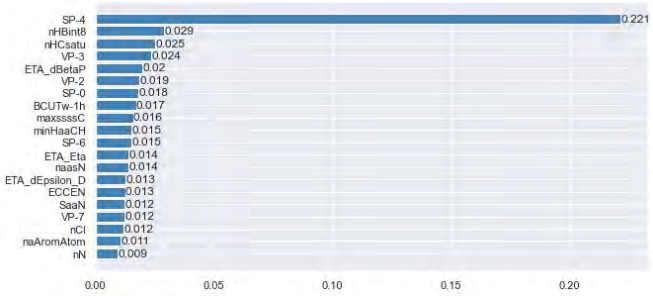
6.6 特征重要性排序和变量关系可视化分析

为进一步观察各个变量对分类模型的影响，本文对各个最优模型进行了特征重要性排序及变量关系的可视化图分析。

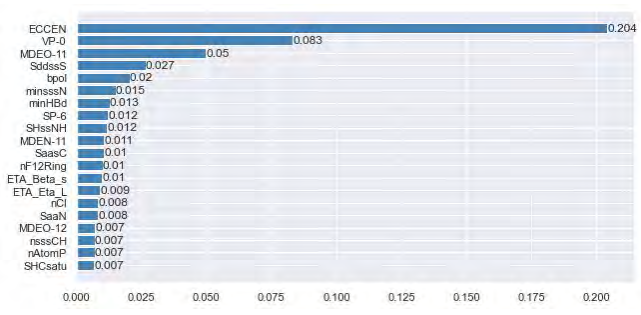
图 6.18 为各个最优模型的特征重要性排序，对于预测 Caco-2 的 XGBoost 模型而言，最为重要的特征是 WPATH，其次为 minaaO，其中 WPATH 重要性远远超过其他特征，对模型准确率的影响极大；对于预测 CYP3A4 的 XGBoost 模型而言，最为重要的特征变量为 SP-4，为 0.221，其次为 nHBint8，接着为 nHCsatu，在该模型下，SP-4 的特征变量将对模型产生极为重要的影响；对于预测 hERG 的 XGBoost 模型而言，最为重要的特征变量是 ECCEN 模型，其次为 VP-0；对于预测 HOB 的 RandomForest 模型而言，最为重要的特征变量为 BCUTc-1l，紧接着为 SsOH、maxsOH、minsOH；对于预测 MN 的 RandomForest 模型而言，最为重要的特征变量为 WTPT-5，接着为 ETA_BetaP_s、TopoPSA 等变量。从特征重要性排序上来看，使用 XGBoost 进行分类预测的模型有时候过于依赖某一个重要变量，而使用 RandomForest 模型进行分类预测，其特征重要性排序值相对而言较为均匀。



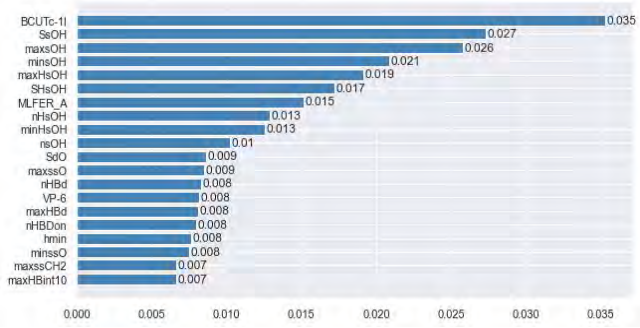
（a）预测 Caco-2 的 XGBoost 特征重要性排序



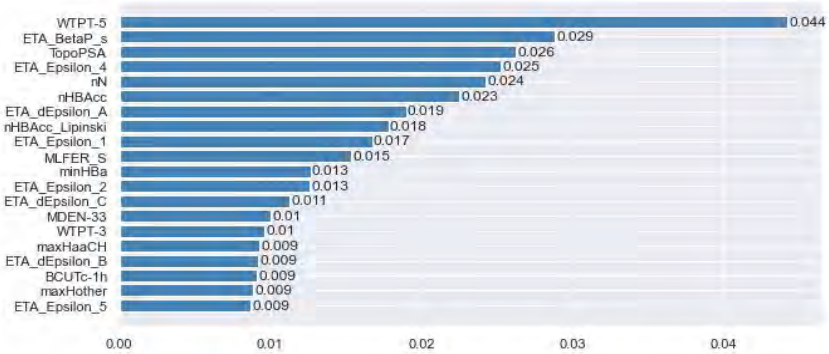
（b）预测 CYP3A4 的 XGBoost 特征重要性排序



（c）预测 hERG 的 XGBoost 特征重要性排序



（d）预测 HOB 的 RandomForest 特征重要性



（e）预测 MN 的 RandomForest 特征重要性

图 6.18 不同 ADMET 性质分类预测最佳模型的特征重要性排序

图 6.19 为影响 HOB 分类前 7 个重要特征变量的关系图，由于图形过多，本文仅选取

影响 HOB 分类前 7 个重要特征变量的关系图进行分析，且仅以 HOB 为例子，左下角为核密度图，从图中可知部分变量对于分类具有重要的区分效果，如 MDEO-11，当其值比较大的时候，HOB 值一般为 0，当其值比较小的时候，HOB 值一般为 1；除此之外，我们可以看到 ECCEN 和 bpol 具有明显的相关关系，bpol 与 VP-0 也具有一定的相关关系；对角线上可以看到在不同变量上 1 和 0 的分布情况，其中 minHBd 图的波动略大；左小角的核密度图展示了二元变量的数据分布情况，可见对于 minHBd 和 minsssN 的二个变量而言，其具有较多的“核”，数据分布较散等。从比较重要的特征数据中，可以看到各个变量数据分布情况及其对分类的影响，以用于指导问题 4 的解决。

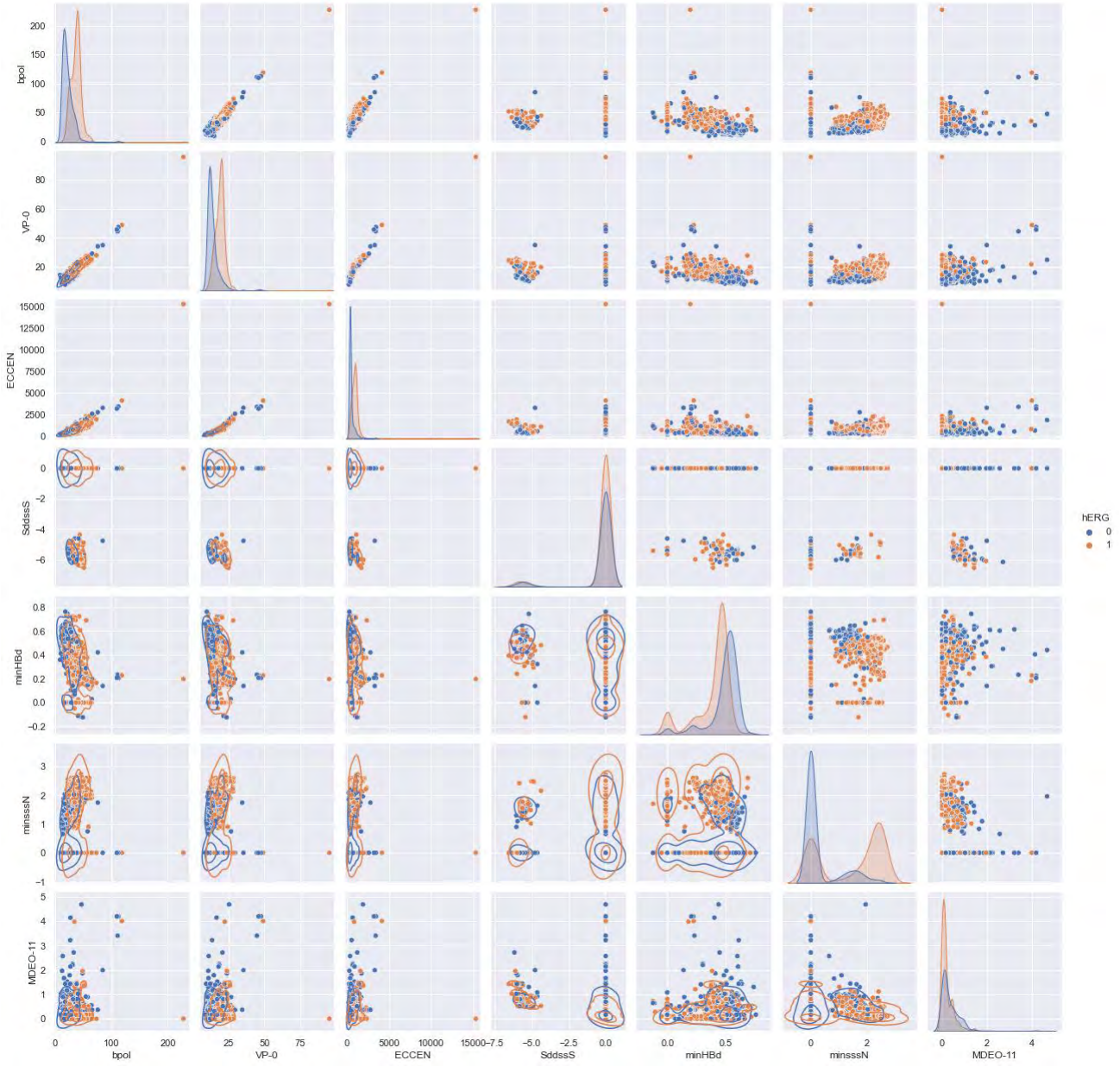


图 6.19 影响 HOB 分类前 7 个重要特征变量的关系图

6.7 模型小结

针对问题 3，本文首先基于证据权重 WOE 和信息值 IV 确定了预测 Caco-2、CYP3A4、hERG、HOB、MN 的重要变量，将这些变量作为模型的输入，其中，预测 Caco-2 性质的分子描述符选择 282 个变量；预测 CYP3A4 性质的分子描述符选择 310 个变量；预测

hERG 性质的分子描述符选择 290 个变量；预测 HOB 性质的分子描述符选择 269 个变量；预测 MN 性质的分子描述符选择 289 个变量。同时对各个 ADMET 性质进行分析，发现 ADMET 性质数据分布不平衡，采取过采样的方法使数据均衡，防止出现过拟合现象。

然后累计选取了 13 个分类模型，其中，11 个机器学习模型，2 个深度学习模型，并使用准确率、精度、召回率、F1 值、PR 曲线、ROC 曲线、AUC 值、对数损失等评价指标评价这些模型的优劣，发现在构建的所有模型中，针对 Caco-2、CYP3A4、hERG 的分类预测的最佳模型为 XGBoost 分类模型，针对 HOB 的分类预测的最佳模型为 RandomForest 分类模型；针对 MN 的分类预测的最佳模型为 XBGoost 分类模型和 RandomForest 分类模型，为了模型多样性，在预测 MN 值的模型选择上，使用 RandomForest。在对各个模型进行评估的时候，发现在该样本数据集上，深度学习模型并不适合用于做分类预测，选择使用机器学习对数据集的 ADMET 性质进行预测相对而言更加合适。

继而，为了获得最佳模型的最优参数，本文对针对各个 ADMET 性质构建的模型进行了参数调优，确定了不同模型的最佳参数，同时，对各个模型进行了泛化能力评估。最终达到预测 Caco-2 的 XGBoost 模型准确率最高可达到 93.9%；预测 CYP3A4 的 XGBoost 模型准确率最高可达到 96.8%；预测 hERG 的 XGBoost 模型准确率最高可达到 92.6%；预测 HOB 的 RandomForest 模型准确率最高可达到 92%；预测 MN 的 RandomForest 模型准确率最高可达到 97.8%。

最后，本文对构建的各个 ADMET 性质最优模型进行特征重要性分析和变量关系的可视化，发现针对不同的 ADMET 性质，其重要的分子描述符具有很大的不同；而变量关系可视化分析，可用于指导问题 4 的建模，确定不同变量对于分类的影响，检验使用算法得到的范围是否在正确范围内。

七、问题四：模型的建立与求解

7.1 问题分析

第四问基于第二问的定量预测模型和第三问的分类预测模型，找到分子描述符对抑制Erα的生物活性和 ADMET 性质的内在关系，并在一定的阈值内进行寻优。由于化合物抑制 Erα的生物活性和 ADMET 性质是相对独立的，因此并不确定化合物对抑制 Erα具有更好的生物活性时同时具有更好的 ADMET 性质，需要在两个目标之间进行权衡，因此可以将选取分子描述符看作一个多目标问题。在确定下来具体的分子描述符之后，还需要让找到分子描述符的取值范围，让其尽可能更好地满足以上两个条件。本文采用粒子群算法和主要目标法来选取分子描述符并获取最优解，并得出分子描述符的取值范围。

7.2 数据预处理

根据问题要求，需要寻找并阐述分子描述符，并找到他们的取值范围，能够使的化合物对对抑制 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）。前面问题解答进行了一些数据清理操作，筛选出了 196 个分子描述符，这里也将其作为数据源对象范围。

问题中的抑制 ERα具有更好的生物活性是指分子描述符所在的化合物具有具有更好的生物活性，其实验测定值为 IC50，值越小对抑制 Erα活性就越有效，因此取 IC_{50} 值的负对数 pIC50 来表示生物活性值，其值越高生物活性越高。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）指的是分子描述符所在的化合物具有更好的 ADMET 性质。由题目所给出的资料，界定五个性质如何为好如表(7.1)。将化合物对应的 CYP3A4、hERG、MN 的值用 1 减去，得到取值 1- CYP3A4、1- hERG、1-MN，便可以得到该化合物的此三个性质是否为好。在给定的五个 ADMET 性质中至少三个性质较好，可依据该表界定为“好”的取值有三个及以上， 即 Caco-2、1- CYP3A4、1- hERG、HOB、1-MN 五个取值相加是否大于等于 3，用公式(7-1)表示某一化合物的 ADMET 性质为:

$$g_{ADMET}(SMILES) = (\text{"Caco} - 2\text{"}) + (1 - \text{"CYP3A4"}) + (1 - \text{"hERG"}) + (\text{"HOB"}) + (1 - \text{"MN"}) \quad (7\text{-}1)$$

表 7.1 ADMET 好的性质界定

ADMET 性质	1 的含义	ADMET 性质好的含义	“好”的取值
Caco-2	代表该化合物的小肠上皮细胞渗透性较好	能够被小肠上皮细胞更好吸收	1
CYP3A4	代表该化合物能够被 CYP3A4 代谢	不能代谢速度太快	0
hERG	代表该化合物具有心脏毒性	不具有心脏毒性	0
HOB	代表该化合物的口服生物利用度较好	口服生物利用度好	1
MN	代表该化合物具有遗传毒性	不具有遗传毒性	0

7.3 模型建立与求解

7.3.1 基于粒子群算法的多目标优化问题

（1）Pareto 最优解集[10]

在多目标优化问题中常常存在两个或者多个目标相互冲突的情况，需要算法进行衡量取舍，通常涉及到最大化或者最小化多个冲突目标函数。可以把多目标优化问题表示为数学公式为：

$$\text{自变量：}\quad X=[x_1,x_2,x_3,\ldots,x_{n-1},x_n]^T$$
$$\text{目标：}\quad F(X)=[f_1(X),f_2(X),\ldots,f_m(X)\rightarrow min]$$
$$\text{约束：}\ a_i\le x_i\le b_i(i=0,1,\ldots,n)$$
$$q_j(X)=0(j=0,1,\ldots,r) \tag{7-2}$$
$$p_k(X)\le 0(k=0,1,\ldots,s)$$
$$X\in\Omega$$

上式中：a_i，b_i为第i个设计变量x_i的上、下限，n为设计变量的个数，r为非上、下限等式约束的个数，s为非上、下限不等式约束的个数，Ω为决策变量可行空间。

在上述多优化问题中，往往存在多个优解，组成一个解集，称为 Pareto 解集（非劣解集）。当考虑两个目标$[f_1(X),f_2(X)]$最小的优化问题时，解集所形成的闭合区域能代表优化问题的所有可行解，边界上往往可以得到更好的解。

（2）粒子群算法（PSO）

粒子群算法[11]是一种精度高、易控制、收敛快，源于鸟群捕食行为研究的进化算法（流程如图 1）。通过设计粒子来模拟鸟类，代表优化问题中可行性的解，每个粒子拥有三个属性——速度、位置和适应度值。每个粒子在搜索空间中单独搜索个体最优解$Pbest$，并将其记为当前个体极值与整个粒子群的粒子共享，找到的最优的那个个体极值将作为整个粒子群的当前全局最优解$Gbest$，所有粒子再通过当前全局最优解和个体极值来调整自己的位置，直到找到满足条件的全局最优解。

粒子维度为D，粒子的数量即种群大小对N，假定种群在第t次迭代产生的第$i(i=1,2,\ldots,N)$个粒子的位置为：

$$present_i(t)=\left(x_{i,1}(t),x_{i,2}(t),x_{i,3}(t),\ldots,x_{i,D}(t)\right) \tag{7-3}$$

那么在第t次迭代产生的第 i 个粒子的速度可以表示为：

$$v_i(t)=\left(v_{i,1}(t),v_{i,2}(t),v_{i,3}(t),\ldots,v_{i,D}(t)\right) \tag{7-4}$$

对于种群在$t+1$ 次迭代时粒子的速度以及位置依据(7-5)和(7-6)两个公式分别进行更新：

$$v_i(t+1)=\omega*v_i(t)+c_1*rand*(Pbest_i-present)+c_2*rand*(Gbest_i-present) \tag{7-5}$$
$$present_i(t+1)=present_i(t)+v_i(t+1) \tag{7-6}$$

上面两式中$Pbest_i$表示第i个粒子历史飞行中的最优位置，$Gbest_i$表示全部粒子历史飞行中的最优位置。ω称为惯性因子，其值为非负，值越大，全局寻优能力越强，局部寻优能力越弱，这个值的引入可以让粒子群算法性能大大提高，针对不同的搜索问题可以调整全局和局部搜索能力。c_1和c_2表示控制粒子探索时的非负数学习因子，取值为[0,1]，增加搜索随机性，$rand$表示随机值。公式(7-5)中分别为记忆项、自身认知项和群体认知项相加，记忆项指的是上一次迭代的速度大小和方向，自身认知项是指当前点指向粒子自身最好点

的一个矢量，表示粒子的动作来源于自己经验部分，群体认知是指一个从当前点指向种群目前最好点的矢量，反映了粒子间的协同合作和知识同享。粒子通过自己和其他粒子的经验来确定下一步运动。

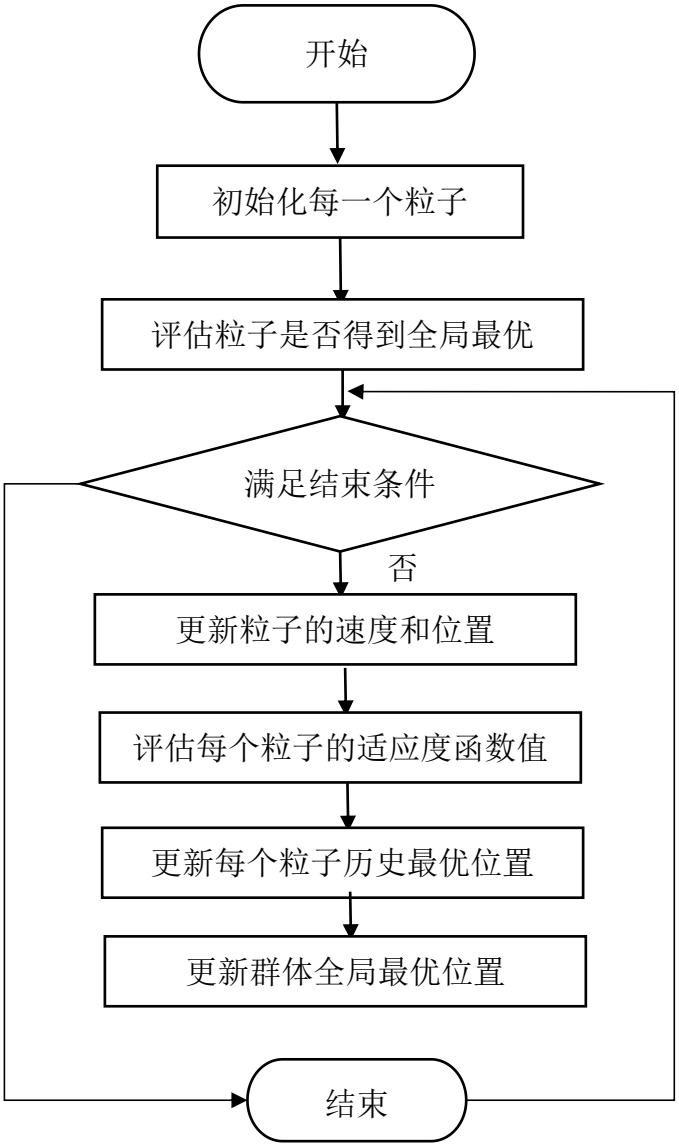


图 7.1　粒子群算法流程图

(3)多目标优化模型

在本文前面解答的问题中，前一二三问已经做了相应的特征重要性排序。第一二问获得了对活性影响大的前二十个分子描述符并得到了回归定量构效模型，由此可以得到分子描述符向量X对抑制 Erα的活性定量函数$f(X)$。第三问则得到了 ADMET 五个分类预测模型，将这五个模型相加也可以得到分子描述符向量 X对 ADMET 总性质的影响函数$g(X)$。第四问依据多目标优化模型所得到的优化模型如(7-7)。

$$find: X = [x_1, x_2, x_3, \ldots, x_{n-1}, x_D]^T$$
$$max: \begin{cases} f_{pIC50} = f(X) \\ g_{ADMET} = g(X) \end{cases} \tag{7-7}$$
$$s.t.: \min(x_i) \le x_i \le \max(x_i), i = 1, \ldots, D$$

上面模型中f_{pIC50}和g_{ADMET}分别表示前三问中得到的所有的分子描述符对生物活性、ADMET 性质的回归函数，分子描述符x_i取值范围仅限于所给出的化合物中的含量范围。在实际操作过程中，基于前三问已经经过了数据清理，剔除了重要性相对低的无效值和缺失值，得到 196 个相对重要的分子描述符，可以将这 196 个分子描述符作为自变量代入模型进行计算。

依据上文介绍的粒子群算法，本文参数设定如表 7.2：

表 7.2 粒子群算法的参数设定

参数	取值（类型）	说明
粒子数量N	10	一般取 10，很小时陷入局部最优，很大时计算难度增加
粒子长度D	196	所定自变量的长度
粒子范围R	$[\min(x_i), \max(x_i)]$	在已知范围内求最优
最大速度V_{max}	$R * 0.15$	一般取为变化范围的 0.1~0.2
社会认知c_1，自我认知c_2	0.8	增加搜索随机性
ω	1	惯性权重
终止条件	迭代次数最大 1000	

（4）模型求解

通过建立以上多目标优化模型，模型求解的过程大致类似于：

步骤 1：初始化，将 10 个粒子的位置和速度进行随机初始化。计算所对应活性函数f_{pIC50}和 ADMET 函数g_{ADMET}的取值（即适应度函数值），得到所有粒子中两函数的最大值，得出个体和群体历史最优位置。

步骤 2：基于设定的参数 R 和V_{max}，运用粒子群算法的迭代步骤，依据当前最优位置和自身位置的评估，得出粒子的速度更新，进而计算位置更新。

步骤 3：评估粒子更新后的适应度函数值，再次选取个体最优解和全局最优解。若满足结束条件，则输出，否则返回步骤 2。

7.3.2 基于主要目标法的优化问题

（1）主要目标法

基于本题并没有太高的复杂性，仅两个目标，约束值也相对简单。因此我们可以选择先验优先权方法——主要目标法，先决策再搜索。此题可以将活性函数作为主要目标函数，而将对 ADMET 的限制条件确定下来作为新的约束条件，这样就可以把多目标优化问题转换为单目标优化问题，目标函数仅为活性函数求最大，模型如(7-8)所示。

$$find: X = [x_1, x_2, x_3, ..., x_{n-1}, x_n]^T$$

$$max: f_{pIC50} = f(X) \tag{7-8}$$

$$s.t.: \begin{cases} \min(x_i) \le x_i \le \max(x_i), i = 1, ..., n \\ g_{ADMET} = g(X) \ge m \end{cases}$$

此式子中上文出现过的符号与模型（7-7）中的含义一致，m表示三个性质好的对应的ADMET 函数对应的值。

（2）模型求解

基于约束条件 ADMET 函数的限制条件，可以得到X的取值范围。限定了取值范围之后，求f_{pIC50}最大值就变成了单目标非线性函数优化问题，可以通过 MATLAB 或者 python 求解。

7.3.3 基于粒子群算法模型的求解结果

基于粒子群算法模型求解部分结果如表 7.3 所示。

表 7.3 粒子群算法模型求解部分结果

分子描述符	取值范围
MDEC-23	12.8~28.5
LipoaffinityIndex	7.5~9.3
maxsOH\ minsOH	9.2~11.8
minssN	1.3~2.1
AMR	121.6~134.3
ETA_Shape_P	0.05~0.2
minaaN	3.7~5.4
minHssNH	0.3~0.4
ATSc3	-0.2~-0.1
hmin	-0.6~-0.1
sumI	45.2~83.9
MLFER_E	1.7~3.4

7.3.4 基于主要目标法的求解结果

基于主要目标法模型求解部分结果如表 7.4 所示。

表 7.4 基于主要目标法的模型求解部分结果

分子描述符	取值范围
hmin	-0.4~-0.1
minHBa	0.1~2.7
ETA_Shape_P	0.03~0.22

BCUTp-1l	1.6~2.3
SwHBa	22.1~33.5
MLFER_E	1.3~2.8
minsssN	1.8~2.3
LipoaffinityIndex	7.2~9.3
suml	54.4~80.6
TopoPSA	44.9~98.1
MDEC-23	13.1~26.4
AMR	111.4~138.6

7.3.5 结果分析

通过以上两个模型求解，求出的部分重叠分子描述符如表 7.5，有些分子描述符分子描述符由于数值相差过大或者由影响并不大可以忽略不计，展示的分子描述符分别为 MLFER_E、AMR、MDEC-23、sumI、LipoaffinityIndex、hmin、ETA_Shape_P、BCUTc-1l，可以观察的到取值范围大部分分布在该分子描述符在所给化合物中含量的平均值附近。

除了模型求解，本文将其中几个分子描述符做了活性和 ADMET 分布的散点图。有图 7.2 和图 7.3 可以看出 MDEC-23 和 LipoaffinityIndex 都随着含量变化，所在化合物的活性和 ADMET 性质都有肉眼可见的变化趋势。这两图也可以验证以上两种算法有效地找到了这几个分子描述符能够使化合物对 ERα具有更好的生物活性，同时具有更好的 ADMET 性质。而图 7.4 是 C2SP2 所在化合物活性和 ADMET 性质的散点图，点数分布相对均匀，因此这个分子描述符的含量对两者的影响是比较小的。

表 7.5 所筛选分子描述符和取值范围

分子描述符	取值范围
MLFER_E	1.7~2.8
AMR	121.6~134.3
MDEC-23	12.8~26.4
sumI	45.2~80.6
LipoaffinityIndex	7.2~9.3
hmin	-0.3~-0.1
ETA_Shape_P	0.03~0.2
minsssN	1.8~2.1

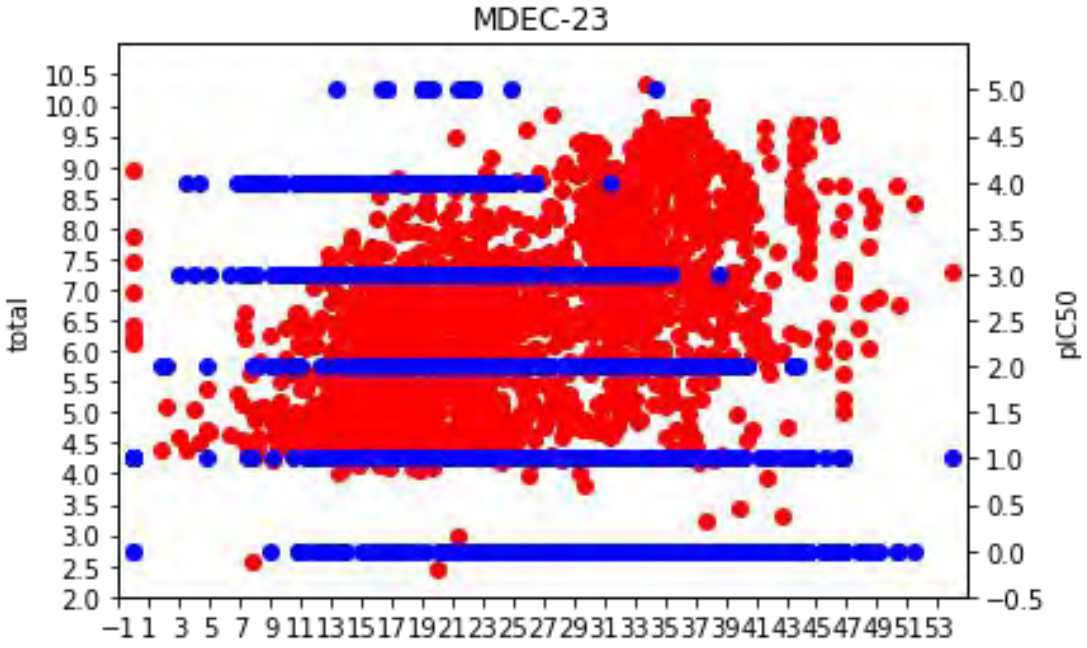


图 7.2 MDEC-23 活性和 ADMET 散点图

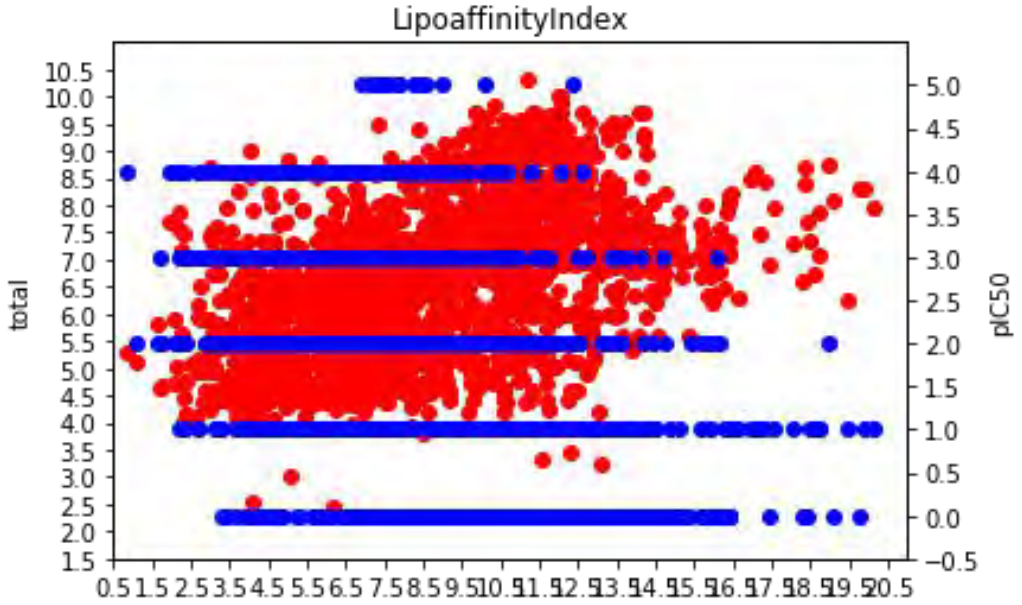


图 7.3 LipoaffinityIndex 活性和 ADMET 散点图

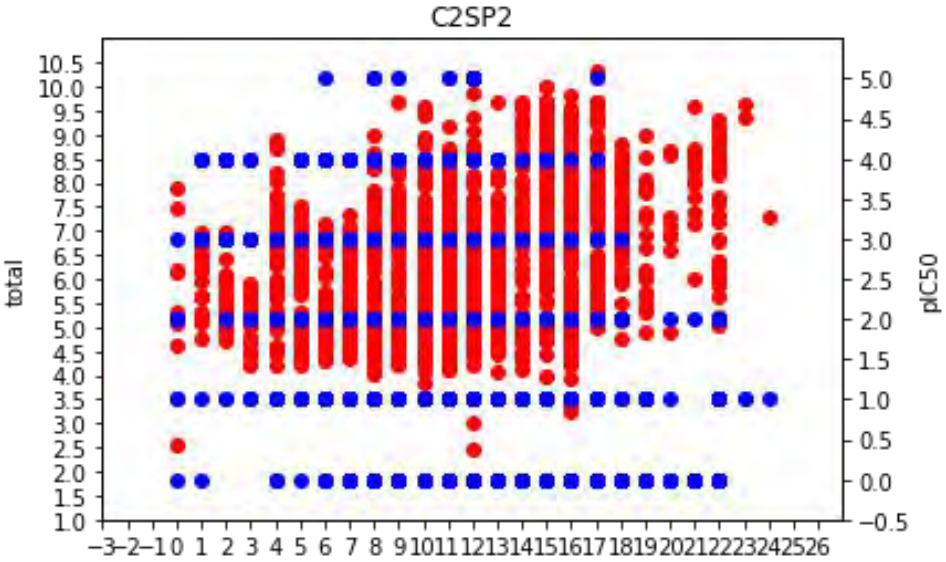


图 7.4　C2SP2 活性和 ADMET 散点图

7.4 模型小结

本文中第四问为多目标优化问题。在进行前三问的数据预处理和模型构建之后，首先基于进化算法中的粒子群算法，通过迭代求出了 Pareto 解集。面对多个目标函数，粒子群算法的优点是所需的参数少，简单易行，收敛速度快。但是经过多次参数的调整，我们观察到粒子群算法在迭代的过程中容易陷入局部最优解，所以参数的选择对算法的性能是比较重要的，本文中的模型还可以继续优化参数。随后为了确保粒子群算法的有效性，还采取了目标规划法将多目标函数转化为单目标函数，用 Python 算出来最优值，可以发现粒子群算法的大部分值都在主要目标法所求的值附近，个别维数可能陷入全局最优解。依据所求得的分子描述符和其取值范围，我们可以更好地优化合成药物 ERα拮抗剂的生物活性和 ADMET 性质。

八、参考文献

[1] 王君瑜,刘雪丽,王海桃,杨海.抑制人乳腺癌细胞 MCF-7 生长的查尔酮类化合物的三维定量构效关系研究[J].中国药房,2016,27(34):4787-4790.
[2] 黄益平,邱晗.大科技信贷：一个新的信用风险管理框架[J].管理世界,2021,37(02):12-21+50+2+16.
[3] 王晓旭,刘晓霞.NOBEL:一种基于拓扑信息与监督学习的蛋白质复合物识别方法[J].中文信息学报,2021,35(09):82-93.
[4] 欧阳志友,陈晨,王愉茜,陈金刚,殷昭,周青松.基于自然语言处理的蛋白质小分子亲和力值预测[J].应用科学学报,2019,37(03):327-335.
[5] 陈金月,王石英.岷江上游生态环境脆弱性评价[J].长江流域资源与环境,2017,26(03):471-479.
[6] Siddiqi, Na ee m. Credit risk scorecards[J]. Wiley, 2012, 10:131-134.
[7] 刘志惠,黄志刚,谢合亮.大数据风控有效吗?——基于统计评分卡与机器学习模型的对比分析[J].统计与信息论坛,2019,34(09):18-26.
[8] 曾子明,万品玉.基于双层注意力和 Bi-LSTM 的公共安全事件微博情感分析[J].情报科学,2019,37(06):23-29.
[9] 周飞燕,金林鹏,董军.卷积神经网络研究综述[J].计算机学报,2017,40(06):1229-1251.
[10] 胡旺,Gary G. YEN,张鑫.基于 Pareto 熵的多目标粒子群优化算法[J].软件学报,2014,25(05):1025-1050.
[11] 王东风,孟丽,赵文杰.基于自适应搜索中心的骨干粒子群算法[J].计算机学报,2016,39(12):2652-2667.
[12] 过晓芳,王宇平.考虑物流服务水平的物流配送规划多目标模型[J].西南交通大学学报,2012,47(05):874-880.

九、附录

程序 1	11种回归模型主要代码

```
#coding=utf-8
import xgboost
from numpy import loadtxt
from xgboost import XGBRegressor
from sklearn.model_selection import train_test_split
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import GridSearchCV,KFold
import shap
import matplotlib
from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score
from math import sqrt
from sklearn import linear_model, tree, svm, neighbors, ensemble,
ExtraTreeRegressor,BaggingRegressor, Lasso
from sklearn.neural_network import MLPRegressor
from sklearn.inspection import permutation_importance
#Linear Regression
model_LinearRegression = linear_model.LinearRegression()
#Decision Tree Regressor
model_DecisionTreeRegressor = tree.DecisionTreeRegressor()
#SVM Regressor
model_SVR = svm.SVR(cache_size=300)
#K Neighbors Regressor
model_KNeighborsRegressor = neighbors.KNeighborsRegressor()
#Random Forest Regressor
model_RandomForestRegressor =
ensemble.RandomForestRegressor(n_estimators=200,random_state=1)
#Adaboost Regressor
model_AdaBoostRegressor = ensemble.AdaBoostRegressor(n_estimators=50)
#Gradient Boosting Random Forest Regressor
model_GradientBoostingRegressor = ensemble.GradientBoostingRegressor(n_estimators=150)
#bagging Regressor
model_BaggingRegressor = BaggingRegressor()
#ExtraTree Regressor
model_ExtraTreeRegressor = ExtraTreeRegressor()
# 载入数据集
df = pd.read_csv('D:/2021 年 中 国 研 究 生 数 学 建 模 竞 赛 赛 题
/Activity_train_1_nounique_nocorr3.csv', encoding='gb18030', header=0)
X = df.drop(["pIC50"], axis=1)#df.drop(["highvalue"], axis=1)
Y = df["pIC50"]
```

```
# 把数据集拆分成训练集和测试集
seed = 7
test_size = 0.25
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=test_size, random_state=seed)
#模型测试
kfold = KFold(n_splits=10)
def gridSearch_vali(model,param_grid,cv=kfold):
    print("parameters:{}".format(param_grid))
    grid_search =
GridSearchCV(estimator=model,param_grid=param_grid,cv=kfold,scoring='neg_mean_squared_error')
    grid_search.fit(X_train,y_train)
    print(grid_search.best_params_)
    return grid_search.best_params_
model =ensemble.RandomForestRegressor(n_estimators=200,random_state=1)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
#特征重要性画图
features = np.array(X_train.columns)
imps_gini=model.feature_importances_
indices = np.argsort(imps_gini)[::-1]
for i in indices:
    print("{0}    {1:.3f}".format(features[i], imps_gini[i]))
perm_importance = permutation_importance(model, X_test, y_test)
indices = np.argsort(perm_importance.importances_mean)[::-1]
for i in indices:
    print("{0}    {1:.3f}".format(features[i], perm_importance.importances_mean[i]))
plt.style.use('ggplot')
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X)
shap_values = explainer(X)
#多预测的解释
shap.summary_plot(shap_values, X, plot_type="bar")
shap.summary_plot(shap_values, X)
HAP=pd.DataFrame(zip(shap_values.feature_names,np.abs(shap_values.values).mean(0)),colu
mns = ['特征','ds重要性'])
shap.force_plot(explainer.expected_value, shap_values[0,:], X.iloc[0,:])
# 对测试集做预测
y_pred = model.predict(X_test)
predictions = [round(value) for value in y_pred]
shap.plots.force(explainer.expected_value,shap_values.values,shap_values.data)
# 评估预测结果
MAE=mean_absolute_error(y_pred, y_test)
MSE=mean_squared_error(y_pred, y_test)
```

```
rmse=sqrt(mean_squared_error(y_test, y_pred))
r2=r2_score(y_test,y_pred)
print(MAE,MSE,rmse,r2)
```

程序 2	证据权重WOE和信息值IV的计算

```
import numpy as np
import pandas as pd
df = pd.read_excel(r'C:/Users/chenm/Desktop/问题3.xlsx','train')
#因变量
Caco2=df['Caco-2']
CYP3A4=df['CYP3A4']
hERG=df['hERG']
HOB=df['HOB']
MN=df['MN']
#自变量/特征值
X = df.iloc[:,6:735]
#卡方分箱、WOE、IV值
def data_vars(df1, target):
stack = traceback.extract_stack()
    filename, lineno, function_name, code = stack[-2]
    #变量名称
    vars_name = re.compile(r'\((.*?)\).*$').search(code).groups()[0]
    #查找变量
final = (re.findall(r"[\w']+", vars_name))[-1]
    x = df1.dtypes.index
    #计算WOE、IV值
count = -1
for i in x:
if i.upper() not in (final.upper()):
if np.issubdtype(df1[i], np.number) and len(Series.unique(df1[i])) > 2:
conv = mono_bin(target, df1[i])
conv["VAR_NAME"] = i
count = count + 1
else:
conv = char_bin(target, df1[i])
conv["VAR_NAME"] = i
count = count + 1
#保存IV结果并返回数据
iv = pd.DataFrame({'IV':iv_df.groupby('VAR_NAME').IV.max()})
iv = iv.reset_index()
    return(iv_df,iv)
```

程序 3	11种机器学习模型建立及评估主要代码

```
#循环分类器构建
for clf in classifiers:
        name = clf.__class__.__name__ #获取分类器名称
        clf.fit(x_train, y_train)#构建训练集上的模型

        scores = cross_val_score(clf, x_train, y_train, cv=10)#十折交叉验证
        scores.mean()

        print("="*35)#模型结果输出模板
        print(name)
        print('****Results****')
        train_prediction = clf.predict(x_test)
        acc = accuracy_score(y_test, train_prediction)
        print("Accuracy: "+str(acc))#准确率计算

        train_predictions = clf.predict_proba(x_test)
        logloss = log_loss(y_test, train_predictions)#logloss结果结算
        print("Log Loss: {}".format(logloss))
        report = classification_report(y_test,train_prediction)
        print(report)
        #混淆矩阵绘制
        cnf_matrixSolo = metrics.confusion_matrix(y_test, train_prediction)
        class_names = [0,1]
        p=sns.heatmap(pd.DataFrame(cnf_matrixSolo,annot=True,
cmap="Blues" ,fmt='g')
        plt.title(name+' Confusion matrix', y=1.1)
        plt.ylabel('Actual label')
        plt.xlabel('Predicted label')
        plt.show()
        #模型评估结果存储，包括fpr,tpr,auc,precision,recall等值
        jilu = pd.DataFrame([[name, acc*100, logloss]], columns=log_cols)
        log = log.append(jilu)
        evaluation.loc[name, 'fpr'], evaluation.loc[name, 'tpr'], thresholds =
metrics.roc_curve(y_test, clf.predict_proba(x_test)[:,1])
        evaluation.loc[name, 'auc'] = metrics.auc(evaluation.loc[name, 'fpr'],
evaluation.loc[name, 'tpr'])
        evaluation.loc[name, 'pre'], evaluation.loc[name, 'rec'], thresholds =
metrics.precision_recall_curve(y_test, clf.predict_proba(x_test)[:,1])
    print("="*30)
```

程序 4	2种深度学习模型建立及评估主要代码

```
#LSTM模型和CNN模型计算评估
#基础数据建构
feanum=730
window=5
features=df1
seq_len=window#窗口数值
amount_of_features = len(features.columns)#数据列数
data = eatures.values#pd.DataFrame(stock)  将表格进行一定的转换
sequence_length = seq_len + 1#增加序列长度
result = []
result = np.array(result)#
row = round(0.8 * result.shape[0])#对数据进行划分
train = result[:int(row), :]
x_train = train[:, :-1]
y_train = train[:, -1][:,-1]
x_test = result[int(row):, :-1]
y_test = result[int(row):, -1][:,-1]
#reshape数据维度
X_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], amount_of_features))
X_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], amount_of_features))
#数据标签转换
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
#LSTM建模
model = Sequential()
model.add(LSTM(16, input_shape=(window, feanum), return_sequences=False))
model.add(Activation('relu'))
model.add(Dense(units=2))
model.add(Activation('softmax'))
model.compile(loss='binary_crossentropy',
                    optimizer=sgd,
                    metrics=['accuracy'])
history=model.fit(X_train, y_train, epochs = 100, batch_size = 64,validation_data=(X_test,
y_test)) #训练模型100次
#CNN建模
model = Sequential()
#一层卷积层，包含了32个卷积核
model.add(Conv1D(32,2, activation='relu', input_shape=(window, feanum)))
model.add(Conv1D(32,2, activation='relu'))
#最大池化层
model.add(MaxPooling1D(pool_size=2))
```

```
#遗忘层
model.add(Dropout(0.25))
#添加卷积层
model.add(Conv1D(64, 1, activation='relu'))
model.add(Conv1D(64, 1, activation='relu'))
#池化层
model.add(MaxPooling1D(pool_size=1))
model.add(Dropout(0.25))
#压平层
model.add(Flatten())
#全连接层
model.add(Dense(256, activation='relu'))
#遗忘层
model.add(Dropout(0.5))
#分类层
model.add(Dense(2, activation='softmax'))

sgd = SGD(lr=0.00001, decay=1e-9, momentum=0.6, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd,metrics=['accuracy'])
history=model.fit(X_train, y_train, epochs = 100, batch_size = 64,validation_data=(X_test,
y_test))
#绘图，绘制ROC曲线和PR曲线。
loss = history.history['loss']
val_loss = history.history['val_loss']
import matplotlib.pyplot as plt
epochs = range(1, len(loss) + 1)
plt.figure(figsize=(12, 4))
plt.title('Training and Test loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
```

程序 5	分类模型参数调优

```
from sklearn.model_selection import cross_val_score

#导入十折交叉验证方法
score_lst = []
best_score = -1

#确定n_estimatorss和max_depths的取值范围，并输出最优的表现，将其输出，然后进行图形绘制，可视化最优参数。
for n_estimatorss in [200,210,220,230,240,250,260,270,280,290,300]:
    for max_depths in [15,16,17,18,19,20,21,22,23,24,25]:
        forest_reg = XGBClassifier(n_estimators = n_estimatorss,bootstrap=False,max_depth = max_depths, random_state=42)
        score = cross_val_score(forest_reg,x_train,y_train,cv=10,scoring='accuracy')
        score_lst.append(score.mean())
        if score.mean() > best_score:
            best_score = score.mean()
            best_parameters = {'n_estimators':n_estimatorss,"max_depth":max_depths}
            print(n_estimatorss,max_depths,score.mean())
        else:
            pass

print('Best socre:{:.2f}'.format(best_score))
print('Best parameters:{}'.format(best_parameters))
print(score_lst)
```

程序 6	粒子群算法

```
%looking for min-functinal value using "Practicle Swarm Optimization"

function [Gbest_x,Gbest_y]=PSO()
%Parameter settings
lower_bound = 0;
higher_bound = 9;
particle = 10;                                  % 粒子个数
max_iteration = 1000;                           % 最大迭代数
dimension = 197;                                % 粒子位置的维度，即自变量个数
c1 = 0.8;                                       % 加速常数1，控制局部最优解
c2 = 0.8;                                       % 加速常数2，控制全局最优解
w = 1;                                          % 惯性因子
vmax = [min(x),max(x);                                  % 速度最大值
precision=0.001;                                % 精度设置
%Initialize
x = lower_bound+rand(particle,dimension).*(higher_bound-lower_bound);
v = 2*rand(particle,dimension);mport numpy as np
Pbest_x=x;                                      % 将初始位置设置为局部最优解的位置
Pbest_y=target(x);                              % 各个粒子的函数值作为其局部最优解
Gbest_y=inf;                                    % 全局最优解的初始值设置为inf
Gbest_x = x(1,:);                               % 初始全局最优位置设定为第一个粒子的
位置
k=1;
f=zeros(particle,1);

%plot the function
horizon = linspace(0,9,500);
vertical = target(horizon);
plot(horizon,vertical);
hold on
[m,index] = min(vertical);
text(horizon(index)+0.5,m,['{F_{min}}= ',num2str(m)])
pic_num = 1;
while k<=max_iteration
    flagx=Gbest_x;
    flagy=Gbest_y;

    % 搜寻各个粒子的局部最优
    for i=1:particle
        f(i) = target(x(i,:));
        if f(i)<Pbest_y(i)
            Pbest_y(i)=f(i);          % Personal best function value
```

```
                Pbest_x(i)=x(i,:);       % Personal best variable
            end
        end
        % 更新全局最优位置及适应值
        [Gbest_y,index] = min(Pbest_y);
Gbest_x = x(index,:);
% 每一次搜寻之后更新粒子的速度及位置
        for n=1:particle
            v(n,:)=w*v(n,:)+c1*rand()*(Pbest_x(n,:)-x(n,:))+c2*rand()*(Gbest_x-x(n,:));

            % 速度越界操作
            for p=1:dimension
                if v(n,p)>vmax
                    v(n,p)=vmax;
                elseif v(n,p)<-vmax
                    v(n,p)=-vmax;
                end
            end
            x(n,:)=x(n,:)+v(n,:);
        End
% 获取动态搜寻过程gif图
        figure(1)
        scatter(Gbest_x,target(Gbest_x))
        hold on
        h1=text(6,-4,['X by TSO=',num2str(Gbest_x)]);
        h2=text(6,-6,['Y by TSO=',num2str(target(Gbest_x))]);
        pause(0.2)
        drawnow;
        F=getframe(gcf);
        I=frame2im(F);
        [I,map]=rgb2ind(I,256);
        if pic_num == 1
            imwrite(I,map,'PSO.gif','gif', 'Loopcount',inf,'DelayTime',0.2);
        else
            imwrite(I,map,'PSO.gif','gif','WriteMode','append','DelayTime',0.2);
        end
        delete(h1);
        delete(h2)
pic_num = pic_num + 1;
end

string=['Min-value by TSO is ',num2str(Gbest_y),', where x= ',num2str(Gbest_x)];
title(string)
```