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	宁波大学
参赛队号	**No.21116460003**
队员姓名	**1.** 赵禹萌
	2. 马涛
	3. 叶建明

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，研究抗乳腺癌候选药物的优化建模，具有重大的现实意义。本文采用距离相关系数、随机森林、XGBoost、遗传算法等数学算法，对数据进行了深度的挖掘，发现了显著影响化合物生物活性的分子描述符，建立了生物活性与分子描述符的量级关系，并构建了化合物 ADMET 性质与分析描述符的定性关联，具体作法如下：

针对问题 1，考虑到数据维度较大，直接进行特征选择存在一定挑战。为此我们在数据标准化的基础上，提出了三次特征筛选方法，首先使用灰色关联度分析，进行第一次特征筛选，得到具有相关性的 70 个变量（见图 4-4）；其次使用距离相关系数，进一步细筛，剩余 37 个相关性较强的变量；最后使用随机森林分析变量之间和变量与化合物活性之间的关联，得到 20 个与生物活性显著相关的重要分子描述符（见表 1）。

针对问题 2，在上一问分析显著性的基础上，我们首先使用最大信息系数法和递归特征消除法分析变量之间的独立性，并将高斯混合模型、改进的随机森林方法作为特征选择的对照方法进行对比，从而选出用于化合物生物活性预测的变量，并对预测特征进行独立性检验；其次考虑到数据样本存在高度的非线性、强耦合、稀疏性、训练样本稀少，深度算法易于造成过拟合问题，我们选取了 XGBoost 算法来学习化合物生物活性与分子描述符之间的定量关系，与随机森林、GDBT、集成学习等进行对比实验；最后在定性与定量的角度进行了可视化展示（见图 5-13 与表 5），直观的证明了本文方法的效果。

针对问题 3，对于 ADMET 五个分类变量，我们在进行数据标准化的基础上，分别使用随机森林对数据进行降维处理，并使用遗传算法选择对其性质具有显著影响的变量。其次使用 XGBoost 对 ADMET 五个性质单独建立分类模型，并设置一组对照实验，以验证本文使用方法的性能。最后我们展示了特征选择模型和 ADMET 性质分类模型的可视化结果，并进行了实验分析（见表 13 与图 6-13）。

针对问题 4，基于第二、三问构建的预测模型，以生物活性最高和 ADMET 性质最好作为目标，以二、三两问筛选出的分子描述符作为决策变量，构建优化

模型。考虑到本优化问题搜索空间大，全局最优解寻找难度大的特点，我们采用了遗传算法对这一优化问题进行求解。由于分子描述符超过一定范围在应用中不具有实际意义，我们利用所给数据的范围来表示分子描述符的取值范围。利用 Python 编程实现，得到了一组分子描述符和其相应的取值，使得化合物对抑制 ERα 具有更好的生物活性，同时具有更好的 ADMET 性质。最后，进行算法对比分析，证明了遗传算法的有效性和优越性（见图 7-3）。

关键字：三次特征筛选方法　递归特征消除法　随机森林　XGBoost　遗传算法　相关性分析　独立性分析

目录

1. 问题重述

根据一项最新数据的显示，2020 年世界乳腺癌新增人数达 226 万，肺癌为 220 万人，乳腺癌正式取代肺癌，成为全球增长的第一大癌症 [1]。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，其中近一半以上的患者在发展中国家。近年来，我国乳腺癌的发病率呈逐年上升趋势，每年有 30 余万女性被诊断出乳腺癌，形式不容乐观。尤其在东南沿海地区和经济发达的大城市，这一现象更为明显。从发病年龄的角度来看，从确诊年龄的角度看，乳腺癌发病呈正态分布，在 20 岁开始上升，在 45-50 岁之间达到峰值，之后逐渐下降。随着治疗方案和方法的更新换代，全球乳腺癌致死病例在减少，但是在我国乡村或者欠发达地区这一数字仍在增加，整体趋势不容乐观。

乳腺癌靶向治疗是在分子水平对其通路靶点设计药物, 通过药物与受体或调节分子结合, 下调受体表达或者活化下游基因, 使得肿瘤细胞凋亡或者抑制其生长 [2]。

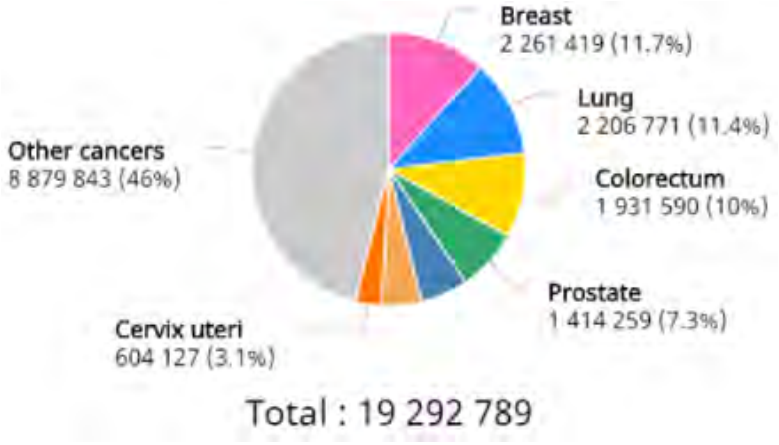


图 1-1　2020 年全球各种癌症占比

目前乳腺癌常见靶向药物有人表皮生长因子受体 (HER) 靶向药物曲妥珠单抗 (rastuzumab)、帕妥珠单抗 (pertuzumab)、西妥昔单抗 (cetuximab) 等。

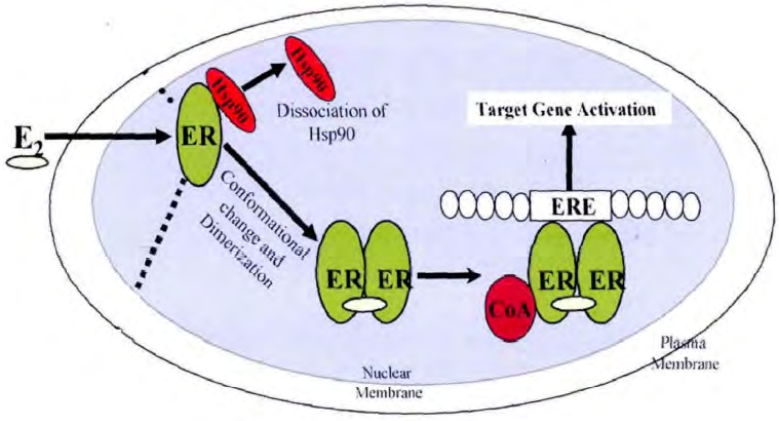


图 1-2　雌激素在机体内发生生物学效应的机制

有研究发现，雌激素受体 α 亚型（Estrogen receptors alpha, ERα）在不超过

10% 的正常乳腺上皮细胞中表达，但大约在 10%-80% 的乳腺肿瘤细胞中表达；而对 ERα 基因缺失小鼠的实验结果表明，ERα 确实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 [3]。除了需要具备良好的生物活性（此处指抗乳腺癌活性）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 [4]。

为此本文旨在解决如下四个问题：

1、对给定数据，探索影响对生物活性最具有显著影响的分子描述符，揭示与活性强相关的分子描述符。

2、构建分子描述符对化合物生物活性的定量预测模型，使用现有的数据对生物活性进行拟合，揭示化合物生物活性与分子描述符的量化关系。

3、根据 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，从而揭示化合物与安全性之间的关系。

4、旨在通过对化合物分子描述符的范围进行优化，使得化合物对抑制 ERα 兼备良好的生物活性和较好的 ADMET 性质。

本文解决问题的思维导图如下所示。

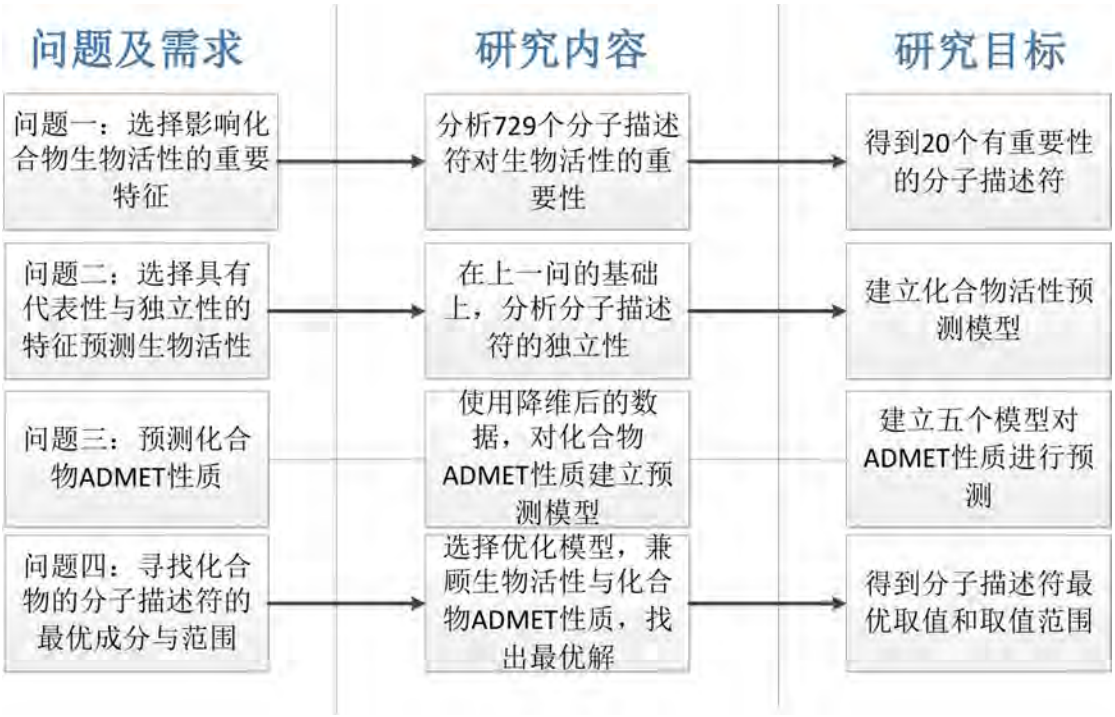


图 1-3　本文解决问题的思维导图

2. 问题假设

(1) 原始样本数据不存在数据属性不完整、数据不一致和噪声数据等问题；

(2) 在优化主要变量时认为所提出的预测模型结果准确；

(3) 化合物分子表达式不同，分子描述符相同的 2D 数据样本在统计学与数据科学中一般不予考虑；

(4) 认为数据中具有手性的同分异构体不会影响后续预测和分类模型；

(5) 未选到的变量对预测模型不会造成影响；

3. 符号说明

序号	符号	含义
1	ER α	雌激素受体 α 亚型
2	ρ	分辨系数
3	$dcorr$	距离相关系数
4	$IMPin$ Gin	基于 Gini 指数的特征重要性
5	$\phi(x/\theta m)$	概率密度函数
6	$\sigma 2$	方差
7	μ	均值
8	μm	均值向量
9	αm	第 m 个高斯分布所占权重
10	Am	第 m 个高斯分布分量的协方差矩阵
11	ωk	节点 k 占总样本的比例
12	$\omega lef\ t$	节点 k 的左节点占总样本的比例
13	$\omega right$	节点 k 的右节点占总样本的比例
14	VC	交叉验证
15	R^2	判定系数
16	$x_1, \ldots, x_{70}$	优化中的分子描述符
17	$Reward$	奖励函数
18	x_{imin}	主要操作变量可取最小值
19	x_{imax}	主要操作变量可取最大值

4. 问题一：模型的建立与求解

4.1 问题分析

问题一要求我们根据文件提供的数据，对生物活性影响的重要性进行排序，并且给出前 20 个对生物活性最具有显著影响的分子描述符。根据题意，此问主要考察对化合物生物活性的相关性影响较大的分子描述符。我们首先详细分析了题目所提供的数据。样本维度大于 700 维，在此情况下直接使用单指标进行降维可能会损失原数据集中的一些信息，因此我们采用了三级特征筛选方法，逐级逐次筛选出强相关的分子描述符。具体来说，我们首先采用灰色关联度分析进行第一次特征筛选，保留 0.7 分数以上，共计 150 个分子描述符样本。我们发现灰色关联分析无法确定分子描述符与化合物的生物活性是否有显著的关联性，只能得到在此规律下指标的得分。所以，本文在第一次结果的基础上，使用距离相关系数再次进行特征筛选，保留 0.25 分数以上的分子描述符。然而上述两次计算，都是在计算分子描述符之间的关联系数，我们需要设计一种面向单变量打分的方法进行完善。为此，我们提出了基于随机森林的分子描述符打分方法，保留 20 个分数最高的分子描述符 [5]。解题思路如下图所示。

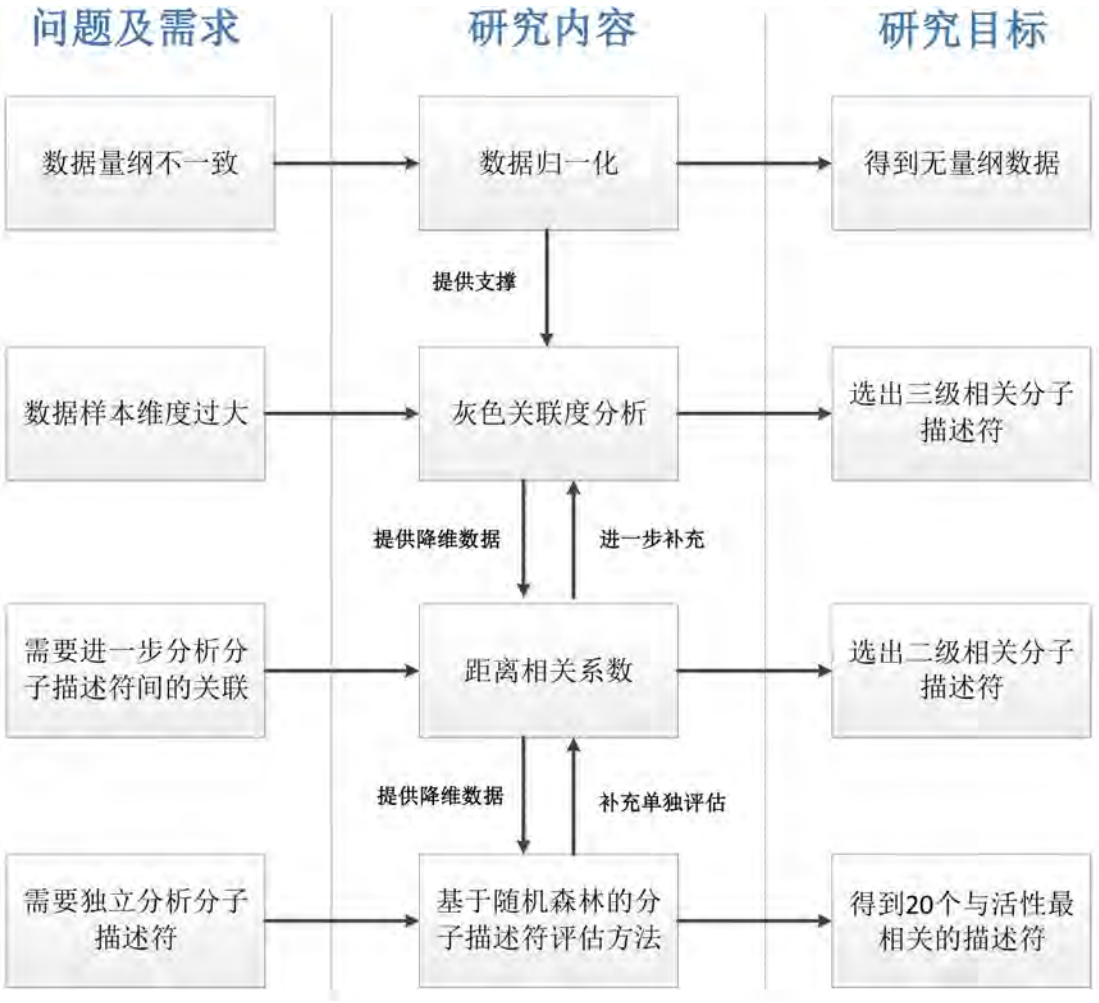


图 4-1　三级特征筛选方法思维导图

三级特征筛选方法流程如下图所示：

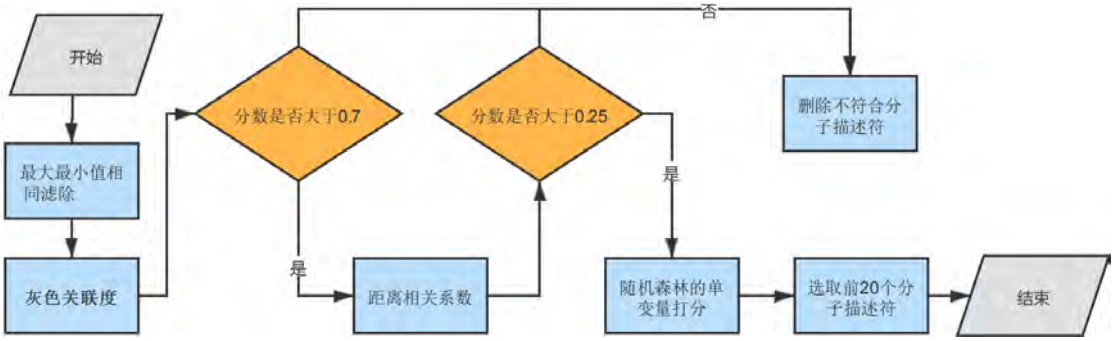


图 4-2　三级特征筛选方法流程

4.2 数据整定

4.2.1 数据分析

我们在分析题目提供的 Molecular-Descriptor 数据时发现，有大量的化合物 Smiles 分子式不同，但是 2D 类型的分子描述符却完全相同，占比 9.2%，分布如下图所示。经查，此类数据属于化学中具有手性的同分异构体，但是在统计学与数学的角度，难以在 2D 分子描述符中进行区分，需使用 3D 类型的描述符进行区分并且此类数据会影响后续模型建立，因此在后续过程中要分析此类数据的影响。在后续模型建立的过程中，我们分别尝试使用全部数据与删除同分异构体的数据分别建模，然而模型建立的效果差别并不大，所以我们在文章中便忽略了此数据的影响，在后续篇幅中便不再提及此问题。

图 4-3　具有手性数据的同分异构体在数据中的分布

4.2.2 数据标准化

由于题目提供数据中包含多类分子描述符，具有独特的含义，所以单位量纲是不同的。为了消除量纲对后续特征提取的影响，我们进行了数据标准化处理，处理方法如下式。

$$x = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (1)$$

其中，x 为标准化后的结果，x_i 为原始数据表中的值，x_{max} 为原始数据表中某分子描述符的最大值，x_{min} 为原始数据表中某分子描述符的最小值。

4.3 基于灰色关联分析的第一次样本筛选方法

4.3.1 算法原理

令化合物的活性评价指标 pIC50 为参考序列 $X_0 = x_0(k), \mathrm{k} = 1, 2, \ldots, \mathrm{n}$, 因素序列 $X_i = x_i(k), k = 1, 2, \ldots, n, i = 1, 2, \ldots, m$, 则 X_0 和 X_i 的灰色关联度 $r(X_0, X_i)$ 定义为:

$$r(X_0, X_i) = \frac{1}{n} \sum_{k=1}^{n} r(X_0(k), X_i(\mathrm{k})) \quad (2)$$

$$r(X_0(k), X_i(\mathrm{k})) = \frac{\min_i \min_k |X_0(k) - X_i(\mathrm{k})| + \rho \max_i \max_k |X_0(k) - X_i(\mathrm{k})|}{|X_0(k) - X_i(\mathrm{k})| + \rho \max_i \max_k |X_0(k), -X_i(\mathrm{k})|} \quad (3)$$

其中, ρ 为分辨系数, 且 $\rho \in [0, 1]$, 其作用在于提高关联系数之间的差异显著性 [6], 通常, $\rho = 0.5$。

4.3.2 计算结果及第一次筛选变量

本文根据灰色关联度模型的相关原理，使用 Python 在 PyCharm 进行了特征选择的操作，对各个分子描述符与化合物生物活性的相关度进行计算。如下图所示。全部结果我们在附件中给出。

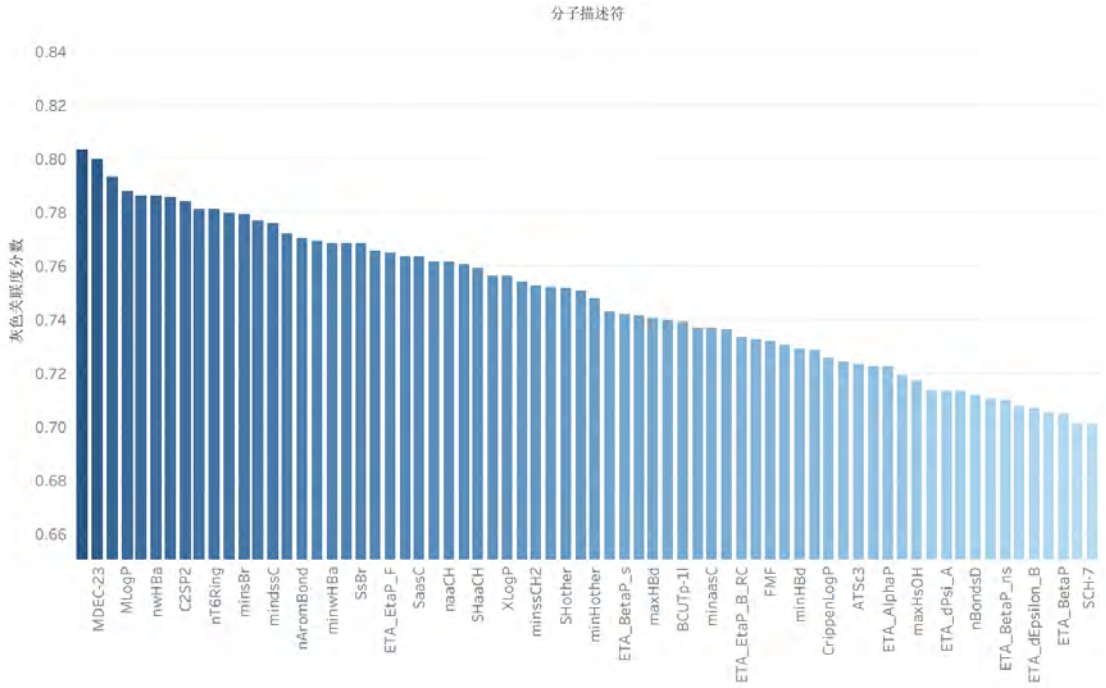


图 4-4　灰色关联度第一次特征筛选

之后我们选取高于 0.7 分数的 70 个分子描述符，作为第一次特征筛选的结果。我们发现灰色关联分析无法确定分子描述符与生物活性是否有显著的关联性只能进行得分的排序，且得到的排序只是在当前灰色规则下的结果。所以我们不能直接使用灰色关联分析直接选出得分前 20 的结果作为问题一的答案。为此我们使用距离相关系数进行第二次特征选择。

4.4 基于距离相关系数的第二次样本筛选方法

在第二次特征选择中，需要在第一次的基础上进一步选出与化合物活性更为为相关的分子描述符。依照题意，我们在本问题中，只需要考虑生物活性与分析描述符的相关性。目前在同类研究中，Pearson 相关系数是最为常用的变量选择方法。Pearson 相关系数默认数据服从正态分布，且只能面向与线性数据。然而，本题所提供数据，具有明显的非线性，因此不能使用 Pearson。为此，我们选择距离相关系数（DISTANCE CORRELATION, DCO）评估化合物生物活性与分子描述符的相关性，并且选择出更为合理的分子描述符。距离相关系数，既可以处理线性数据，也可以处理非线性数据，对数据分布也没有明确的要求。

4.4.1 算法原理

为此我们使用距离相关系数进行来再次筛选，得到 37 个候选分子描述符。在本文中，衡量两分析描述符 x_1 和 x_2 的独立性，记为 $dcorr(x_1, x_2)$. 当 $dcorr(x_1, x_2) = 0$ 时，说明 u 和 v 相互独立；$dcorr(x_1x_2)$ 越大，说明 x_1 和 x_2 的距离相关性越强 [7]。设 $\{(x_i, x_i), i = 1, 2, \cdots, n\}$ 是总体 X 的随机样本，Székely 等 (2008) 定义两随机变量的 x_1 和 x_2 的距离相关系数样本估计值为：

$$dcorr(u, v) = \frac{d\,\mathrm{cov}(u, v)}{\sqrt{d\,\mathrm{cov}(u, u) d\,\mathrm{cov}(v, v)}} \quad (4)$$

4.4.2 计算结果及第二次筛选变量

本文根据距离相关系数模型的相关原理，使用 Python 在 PyCharm 进行了特征选择的操作，对各个分子描述符与化合物活性的距离相关系数进行计算，如下图所示。

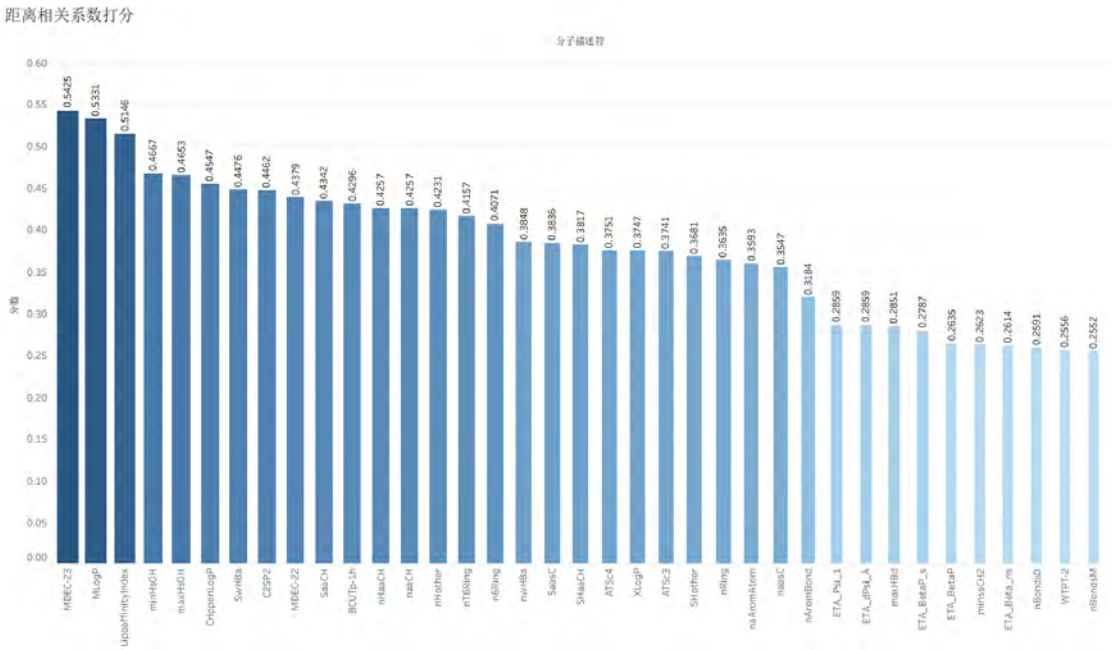


图 4-5　距离相关系数第二次特征筛选方法

之后我们选取高于 0.25 分数的分子描述符，作为第二次特征筛选的结果。然而上述两次计算，都是在计算分子描述符之间的关联系数，我们需要设计一种面向单变量打分的方法进行完善。

4.5 基于随机森林的第三次样本筛选方法

经过上两次的特征选择，数据维度大大降低。此时数据样本数量远大于数据维度，这时便可以使用机器学习的方法进行特征选择。考虑到此处需要对分子描述符进行单变量打分，我们使用随机森林算法对分子描述符逐一打分。

4.5.1 随机森林模型原理

在构建随机森林模型进行特征选择的过程中，我们最终会保留 20 个最显著的分子描述符。我们使用 Gini 系数对分子描述符进行评分，即每一个描述符都被独立的作为自变量去预测化合物生物活性，因此分子描述符的重要性就在此过程中体现出来。基于随机森林的分子描述符打分方法的流程如下图所示。

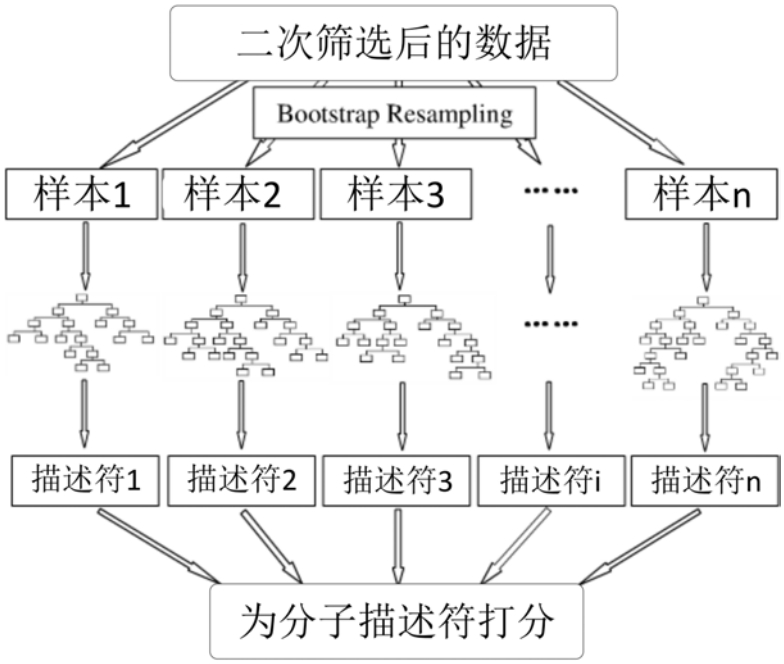


图 4-6　基于随机森林的分子描述符打分方法的流程如图

统计量 $IMP_i^{C\,\mathrm{ini}}$, 即基于 Gini 指数的特征重要性, 表示第 i 个分子描述符在随机森林所有决策树节点上 Gini 指数的平均改变量。

分子描述符 x_i 在节点 n 上的重要性, 即节点 n 上的数据划分到其左右子节点 n_1 和 n_{r} 前后的 Gini 指数变化量如下式所示:

$$IMP_{in}^{\mathrm{Gini}} = I_G(n) - I_G(n_1) - I_G(n_{\mathrm{r}}) \quad (5)$$

若分子描述符 x_i 在第 k 棵决策树中作为节点分割出现的节点集合为 N, 则该分子描述符在这棵决策树上的重要性可由下式得出:

$$IMP_{i\_k}^{\mathrm{Gini}} = \sum_{n \in N} IMP_{in}^{\mathrm{Gini}} \quad (6)$$

若随机森林中有 K 棵树, 则特征 x_i 在整个随机森林中的重要性可由下式计算得出:

$$IMP_i^{\mathrm{Gini}} = \frac{1}{K} \sum_{k=1}^{K} IMP_{i\_k}^{\mathrm{Gini}} \quad (7)$$

4.5.2 计算结果及第三次筛选变量

本文使用上述方法，使用随机森林算法对分子描述符分别计算其重要程度，最终我们保留前 20 个对生物活性最具有显著影响的分子描述符，如下所示。并将其作为本问的答案。

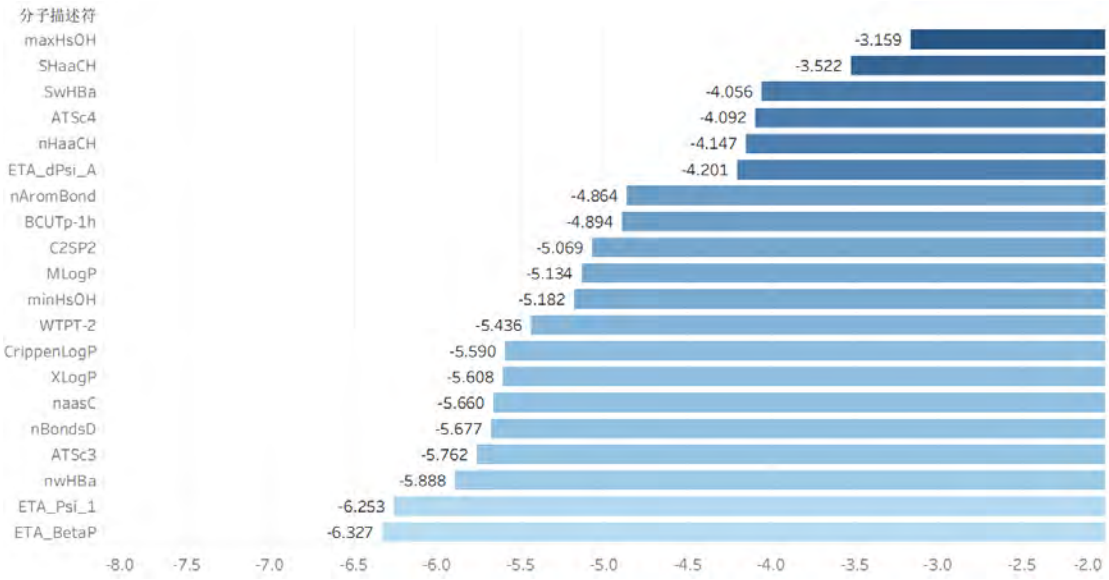


图 4-7　随机森林第三次筛选变量

表 1　问题一答案

序号	分子描述符
1	maxHsOH
2	SHaaCH
3	SwHBa
4	ATSc4
5	nHaaCH
6	ETA_dPsi_A
7	nAromBond
8	BCUTp-1h
9	C2SP2
10	MLogP
11	minHsOH
12	WTPT-2
13	CrippenLogP
14	XLogP
15	naasC
16	nBondsD
17	ATSc3
18	nwHBa
19	ETA_Psi_1
20	ETA_BetaP

5. 问题二：模型的建立与求解

5.1 问题分析

问题 2 旨在通过选择不超过 20 个分子描述符变量，构建分子描述符对化合物生物活性的定量预测模型。首先在此问中，我们需要明确用于构建预测生物活性预测模型的分子描述符，必须要具有代表性与独立性。在问题一中，我们主要分析分子描述符对化合物生物活性的重要性，缺少分析其中的独立性，不能直接将此结果作为化合物活性预测模型的自变量。因此我们在上问第一次特征选择的基础上，着重考虑分子描述符的独立性。为了选择出具有独立性的模型特征，我们分别提出了 1、最大信息系数法 + 递归特征消除法；2、距离相关系数法 + 高斯混合聚类模型；3、改进的随机森林评分法。三种方法选出具有代表性和独立性的特征，从而进行对化合物生物活性定量预测。在这 3 种方法中，距离相关系数旨在对分子描述符进行重要性排序，递归特征消除法和高斯混合模型用于筛选具有代表性的分子描述符，改进的随机森林评分法可以重要性排序的基础上，保留独立性较强的分子描述符。之后我们使用随机森林算法对三种方法选出的特征进行建模预测化合物的活性。我们发现距离相关系数法 + 递归特征消除法选出的分子描述符，并使用极度梯度下降算法预测化合物生物活性取得了最好的预测结果，拟合优度 R^2 可达 79%。

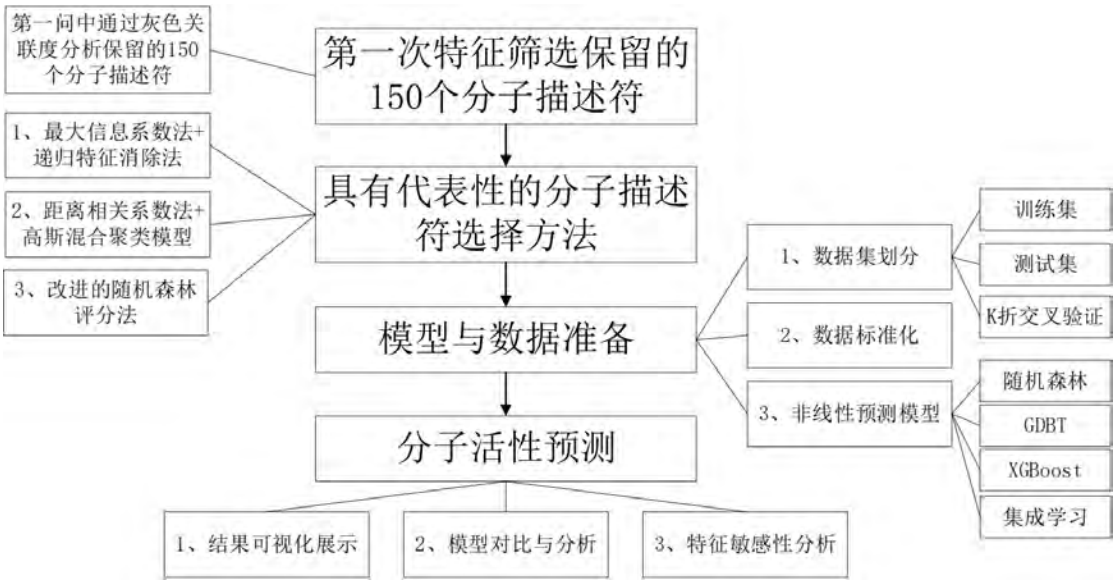


图 5-1　问题二的解题流程

我们进行方法验证的示意图，如下所示。

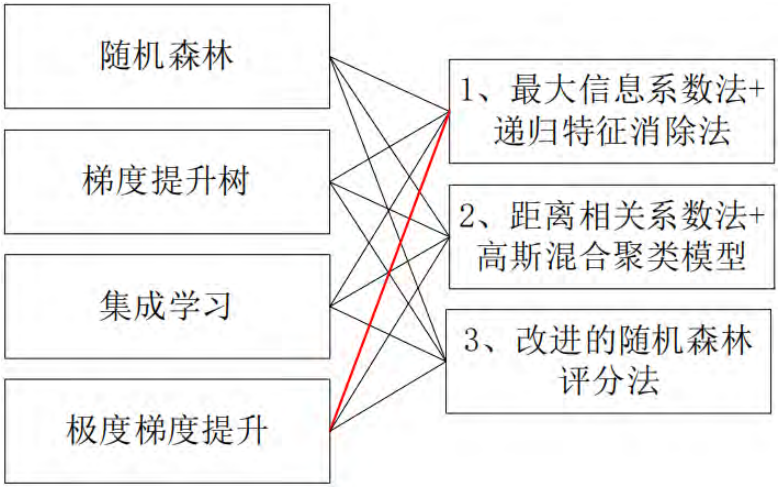


图 5-2　问题二的解题流程

其中共有 3 种特征选择算法，4 种预测模型，我们将其全部进行组合验证，得到 12 组结果，并且发现相关系数法 + 递归特征消除法与极限梯度提升算法结合效果最好，在图中我们使用红线进行了。

5.2 独立性特征选择

5.2.1 最大信息系数法 + 递归特征消除法

最大信息系数法（Maximal Information Coefficient,MIC）一般用于反映自变量和因变量之间的线性与非线性关系，具有较为广泛的应用。在本文中，我们使用最大信息系数法衡量生物活性与分析描述符之间的关联。其计算方式如下所述：

$$I(x;y)=\int p(x,y)\log_2\frac{p(x,y)}{p(x)p(y)}\mathrm{d}x\,\mathrm{d}y \quad (8)$$

$$I[x;y]\approx I[X;Y]=\sum_{X,Y}p(X,Y)\log_2\frac{p(X,Y)}{p(X)p(Y)} \quad (9)$$

$p(x,y)$ 是分子描述符 x 与化合物生物活性 y 的联合概率密度，然而在实际应用中计算联合概率密度具有一定挑战。为此 MIC 的思路是将 x,y 之间的关系，映射到二维空间中，并以散点的形式进行表现，之后将二维空间划分为若干个网格结构。这样就将求解联合概率密度的问题转化为散点在网格中分布的概率。MIC 计算的方法如下所示：

$$\mathrm{MIC}(x;y)=\max_{a*b<B}\frac{I(x;y)}{\log_2\min(a,b)} \quad (10)$$

$$MIC[x;y]=\max_{|X||Y|<B}\frac{I[X;Y]}{\log_2(\min(|X|,|Y|))} \quad (11)$$

我们使用 MIC 对 150 个分子描述符进行打分，我们保留分数大于 0.3 的分子描述符。之后我们使用递归特征消除法（Recursive feature elimination,RFE）进一步选择具有代表性的分子描述符。RFE 的核心思想是通过嵌套一个模型并不断地构建模型，从而判断该分子描述符是保留还是遗弃，然后在剩余的特征上反复迭代，直至所有描述符均被遍历，才终止计算 [9]。在本题中，该特征即是分

析描述符。RFE 算法的稳定性依赖于嵌套模型的选择，本文将随机森林算法嵌套在 RFE 算法中，并进行不断迭代，从而选出具有代表性的分子描述符。

本文使用 RFE 算法提取出 20 个具有独立性的分子描述符，提取效果如下图所示：

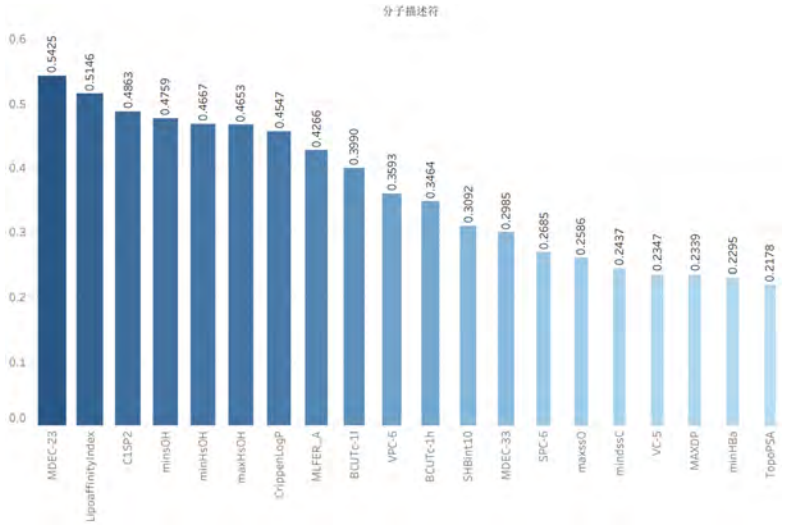


图 5-3　MIC+RFE 方案最后用于预测的分子描述符

5.2.2 距离相关系数法 + 高斯混合模型

在本方案中，我们首先使用距离相关系数法进一步提取重要性较强的分子描述符，再通过使用高斯混合模型（Gaussian Mixture Model，GMM）对样本进行聚类，并选出具有代表性的分子描述符，流程如下所示。

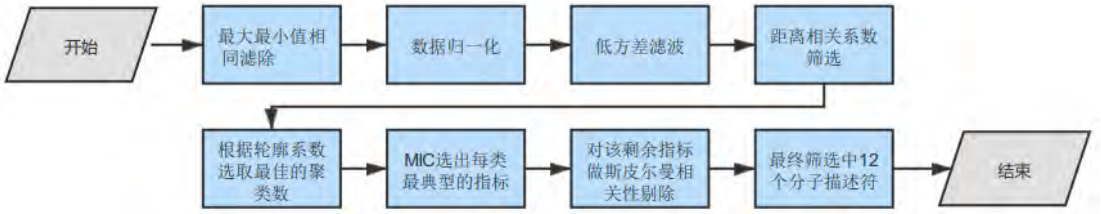


图 5-4　此方案流程

首先我们进行低方差滤波，经过对题目提供数据样本的统计。我们发现方差为 0 的数据条目占比 30.86%，所包含的信息量非常少，需要排除这些因素的干扰。之后计算分子描述符与化合物生物活性的距离相关系数。在第一问的解题中，我们已经介绍过距离相关系数的计算方法，在此不再赘述。我们保留距离相关系数大于 0.2 的分子描述符。

之后我们使用高斯混合模型分析分子描述符的独立性。高斯混合模型望文生义，是以高斯模型为基础的扩展方法，可以将其理解为若干个高斯分布函数（Gaussian Distribution Function，GDF）的线性组合。与高斯函数相似的是，高斯混合模型初始认为所有的数据样本均服从高斯分布，从而计算样本数据集的密度。因此具有相同高斯分布的样本，就被认为是一个簇。通过反复迭代最终计算出混合高斯模型的聚类结果。我们可以将其理解为，用若干个高斯函数对样本数据集进行量化，从而得到最好的分类结果 [10]。

对于一个一维随机变量 X, 服从均值为 μ ，方差为 σ^2 的概率分布，其概率密度函数其概率密度函数为：

$$\mathrm{f}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (12)$$

GMM 是一个软聚类过程，这种聚类方式将样本数据通过概率的大小分属于各个簇，而不是完全地属于某一个簇。

设随机变量为 X, 且混合模型由 M 个高斯分布成员组成，高斯混合模型可以下式：

$$P(x \mid \theta) = \sum_{\mathrm{m}=1}^{M} \alpha_m \phi\left(x \mid \theta_{\mathrm{m}}\right) \quad (13)$$

其中参数 α_m 表示第 m 个高斯分布在混合高斯模型中所占的权重，这个参数一般都是未知的，且满足：

$$\sum_{m=1}^{M} \alpha_m = 1, \quad (\alpha_m \geq 0) \quad (14)$$

虽然通常都不会知晓各个高斯分布在混合模型所占分量，但是由于高斯混合模型是由 M 个高斯分布组合而成的，因此 M 个高斯分布的权重和为 1。

$\phi(x|\theta_m)$ 为混合模型第 m 个分模型的概率密度函数，而最常用的就是高斯密度函数。同时 θ_m 表示为：$\theta_m = (\mu_m, A_m)$ ，μ_m 表示第 m 个高斯分布分量的均值，A_m 表示第 m 个高斯分布分量的协方差矩阵。$\phi(x|\theta_m)$ 可以通过下式计算:

$$\phi\left(x \mid \theta_m\right) = \frac{1}{\sqrt{2\pi A_m}} \exp\left(-\frac{\left(x-\mu_m\right)^2}{2A_m}\right) \quad (15)$$

由以上分析得知一个高斯模型可通过概率密度函数来确定，而概率密度函数主要是由均值 μ 和方差 σ^2 来决定的。故本文将描述混合高斯模型的三个参数记为 $\theta\,\mathrm{m}$ ，即混合系数 α_m，均值 μ_m，协方差矩阵 A_m 。而若需要将样本数据集进行划分聚类成若干个簇，那就需要知道在这个数据集中，哪部分数据服从参数为何值的高斯分布，并且使得聚类结果尽可能地拟合观测数据。其中，在混合模型中的观测数据指的是在数据集中已知某些数据服从已知的高斯分布的那部分数据。同时样本数据集也被称为完整数据，包含着观测到的随机样本即观测数据 $X = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_N\}$ 和未观测到的随机变量即隐含变量 $Z = \{z_1, z_2, \ldots, z_N\}$

由此可见，高斯混合模型主要是通过参数集 θ 决定的。并且为了得到一个较高质量的聚类结果，需要使聚类的簇无限地拟合样本数据集的分布情况。所以需要求解出这个最优的样本参数，最常用找出混合模型参数的方法就是极大化混合模型的对数似然函数。其中，混合模型样本数据集的似然估计函数可通过下式表示:

$$L(\theta) = \mathrm{L}\left(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_m; \theta\right) = \prod_{i=1}^{m} p\left(\mathrm{x}_i, \theta\right) \quad (16)$$

其中 $\mathrm{x}_1 \ldots \mathrm{x}_m$ 来自混合模型样本数据，$p(x;\theta)$ 为模型概率函数,θ 表示一个未知参数或者表示若干个未知参数组成的参数向量。

需要估计的参数的是均值向量 μ_m、协方差矩阵 A_m 以及第 m 个高斯分量所占的权重 α_m。为了较准确估计混合模型这三个参数，会因为仅仅已知部分的高斯分布，在样本数据集里还包含着未观测到的隐含变量 $Z = \{\mathrm{Z}_1, \mathrm{Z}_2, \ldots, \mathrm{Z}_N\}$，若

直接通过求解似然估计函数的最大值来估计混合模型的参数，会使得参数的求解过程十分繁琐复杂，并且不容易进行优化。所以需要先找到隐含变量，当隐含变量确定后，似然函数的最大值就容易确定了，然后建立似然函数对数的下界，同时不断对其下界进行优化。求解参数问题是也为了解决概率模型中存在隐含变量的优化问题，隐含变量计算方法如下式：

$$\mathrm{L}^{(1)}(\theta)=\sum_{i=1}^{m}\log p\left(x_i,\theta\right)=\sum_{i=1}^{m}\log\sum_{z_{\mathrm{i}}}p\left(x_i,z_{\mathrm{i}},\theta\right)\quad(17)$$

GMM 对此数据聚类的轮廓图如下所示。

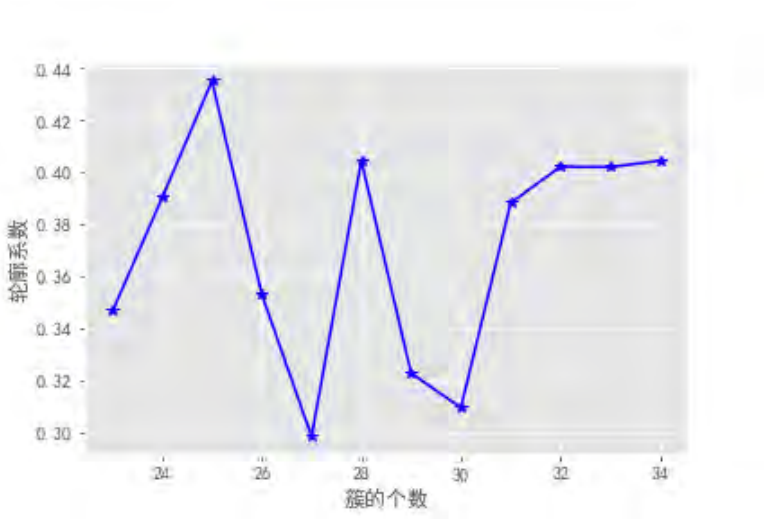
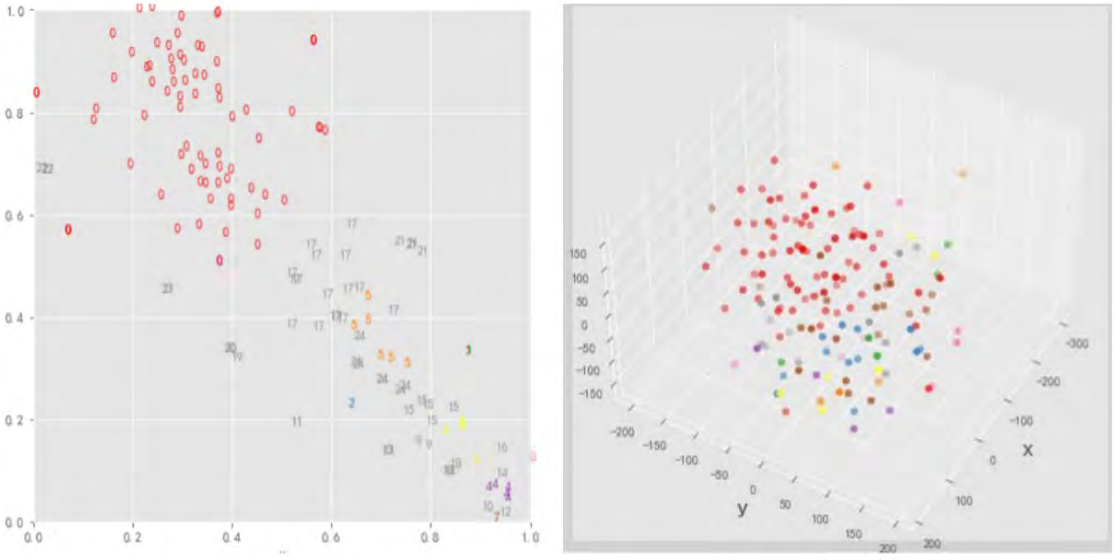


图 5-5　GMM 聚类的轮廓系数

依图所示，轮廓系数在选 24 时，聚类的效果最佳。使用 GMM 模型会给我们选出 24 类分子描述符，我们使用 t-SNE 为聚类结果进行可视化展示，如下图所示：



A). 二维聚类可视化　　B). 三维聚类可视化

图 5-6　GMM 聚类结果可视化

我们使用 MIC 在每一类中选择最有代表性的分子描述符。依照题意要求我们将描述符控制在 20 个以内，因此我们再次使用斯皮尔曼相关系数，剔除相关性较大的的分子描述符。之后我们通过绘制 seaborn 图来检验剩余变量的线性关系，如下图所示：

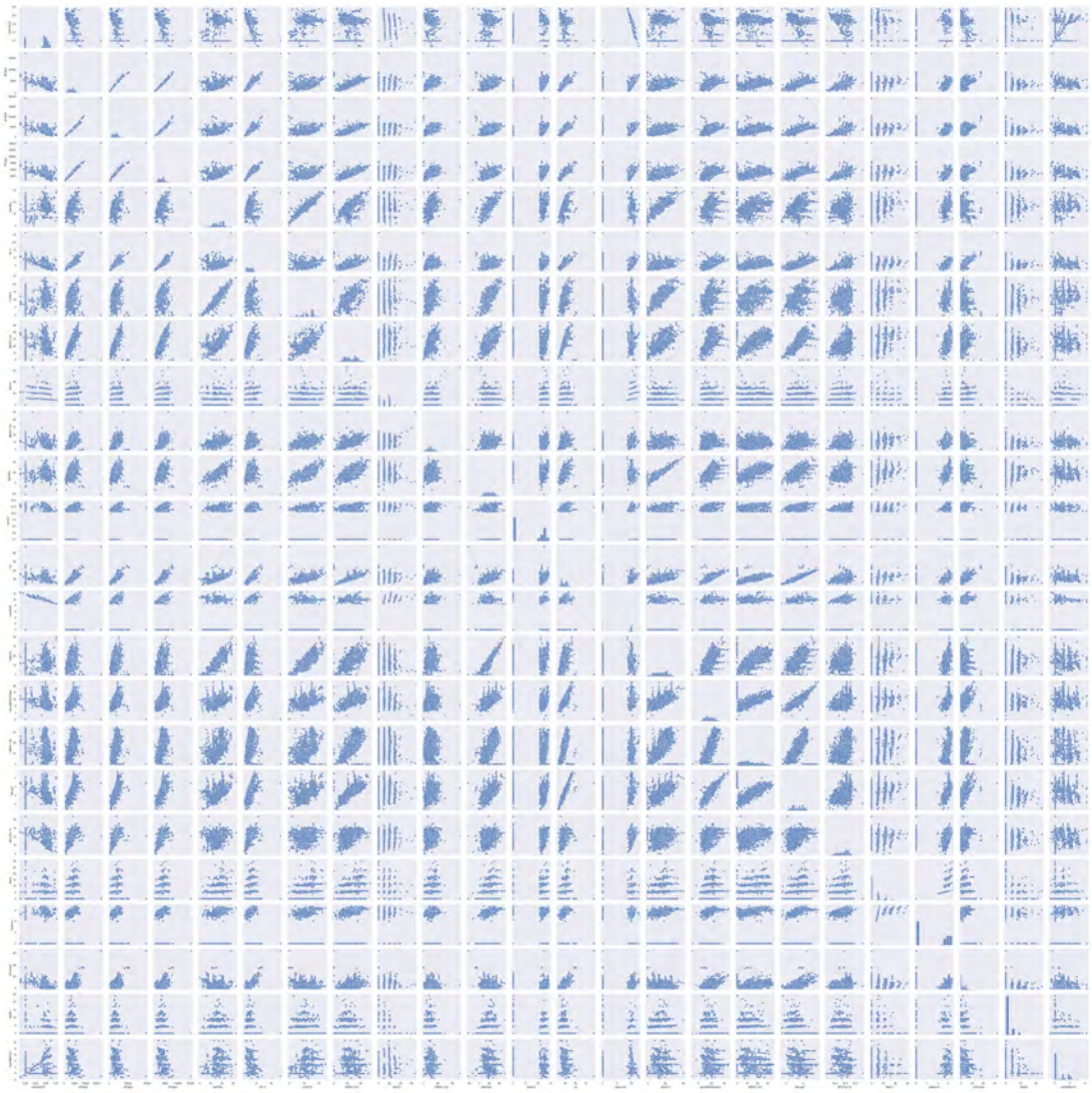


图 5-7　seaborn 非线性检验

图中剩余分子描述符中存在明显的非线性关系，考虑到题目所给数据中的分子描述符与生物活性之间存在明显的非线性，所以我们选取斯皮尔曼相关性系数进行特征筛选。其计算方法如下。

$$\rho = 1 - \frac{6\sum d_i^2}{n\left(n^2-1\right)} \quad (18)$$

我们使用斯皮尔曼绘制分子描述符的相关性热力图，从而科学的剔除相关性较强的特征，如下图所示。

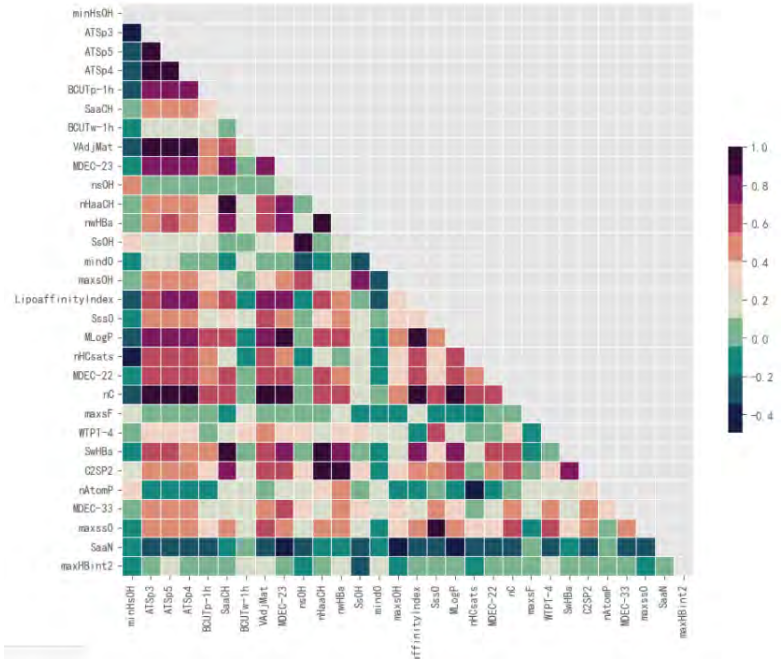


图 5-8　seaborn 非线性检验

最后得到了 12 个分子描述符，如下图所示：

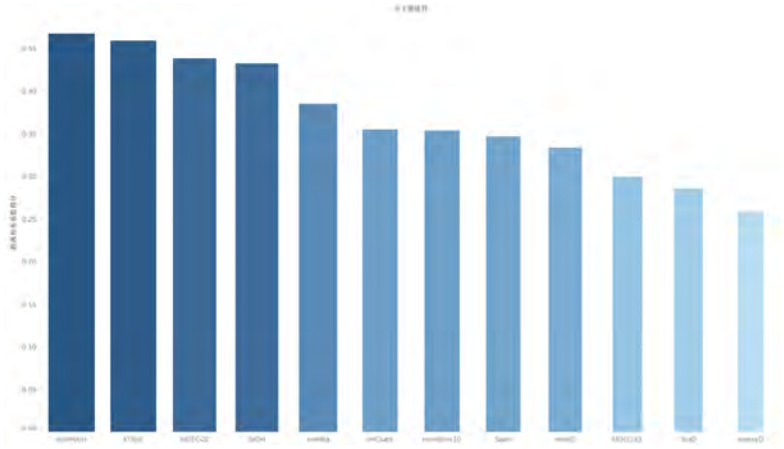


图 5-9　Dco+GMM 方案最后用于预测的分子描述符

5.2.3 改进的随机森林评分法

随机森林算法可以在分类的基础上进行回归分析，通过将样本分类的结果进行一定的运算可以获得各个特征重要性特征的重要性表示特征对预测结果影响程度，某一特征重要性越大，表明该特征对预测结果的影响越大，重要性越小，表明该特征对预测结果越小 [8]。随机森林算法中某一特征的重要性，是该特征在内部所有决策树重要性的平均值，而在决策树中，计算某一个特征的重要性可以采用以下方法：

$$n_k = \omega_k * G_k - \omega_{left} * G_{left} - \omega_{right} * G_{right} \quad (19)$$

其中, $\omega_k, \omega_{left}, \omega_{right}$ 分别为节点 k 以及其左右节点中训练样本与总训练样本数目的比例, $G_k, G_{\text{left}}, G_{right}$ 分别为节点 k 以及其左右子节点的不纯度。知道每

一个节点的重要性之后，即通过公式得出某一特征的重要性。

$$f_i = \frac{\sum_{j \in nodes \text{ split on feature } i} n_j}{\sum_{k \in \text{ all nodes}} n_k} \quad (20)$$

然而上述过程没有考虑，变量之间的相关性，这样的计算效果并不能为预测模型选出最优的特征，为此我们提出了改进的随机森林打分机制，在这个结果的基础上，融合了相关性计算，如下式。

$$f_{ni} = \frac{f_i}{\sum_{j \in \text{ all features}} f_j} * (1 - \frac{6 \sum d_i^2}{n\,(n^2 - 1)}) \quad (21)$$

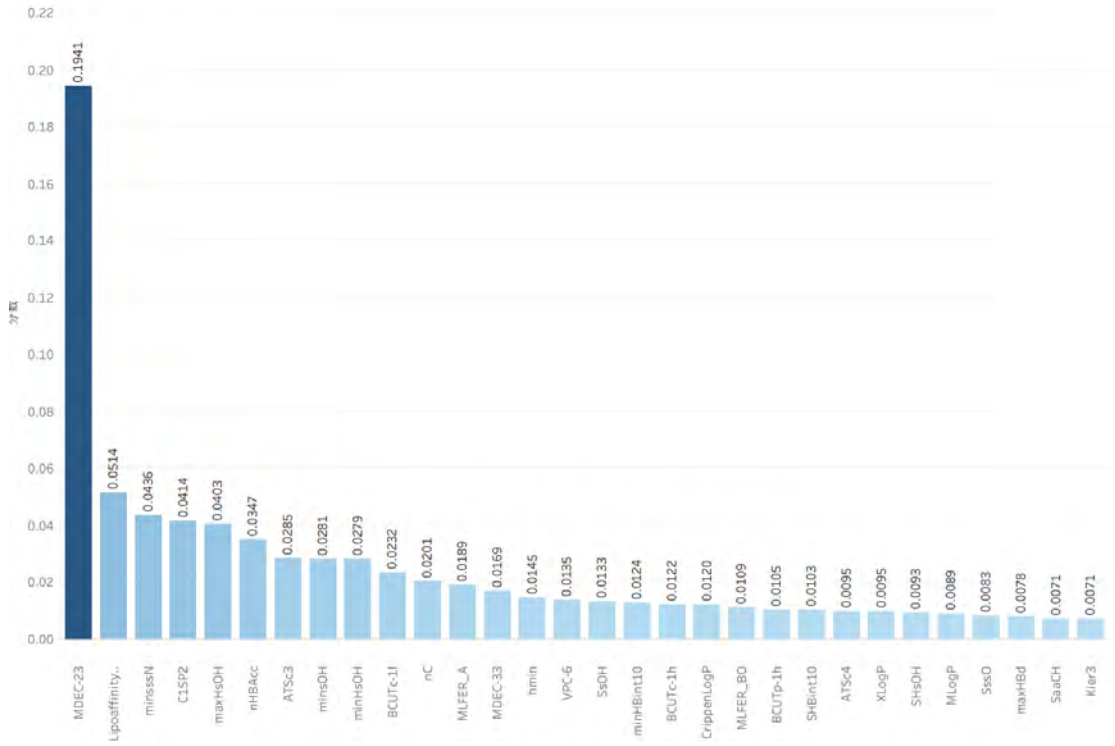


图 5-10　改进的随机森林方案选出的分子描述符

5.3 数据准备

5.3.1 数据集的划分

我们使用三种方法进行了分析描述符选取，得到了三种结果，所以在数据预测阶段，要对三种结果一起进行预测。并且选出最优的方案，当做问题的答案。因此在进行预测之前，需要对三种结果进行数据集划分。在使用模型预测生物活性之前，我们按照 4:1 对数据集进行划分，并且使用 K 折交叉验证，最后使用 R^2 来评估模型的预测能力。因为在处理上一问时，我们已经进行了数据归一化，此问基于第一问的结果，所以在此不再进行数据归一化。

5.4 模型的构建

在此问中我们使用随机森林、GDBT、极端梯度提升、集成学习分别对化合物的生物活性进行预测，以实现找到最优的解决方案。

5.4.1 随机森林

随机森林处理流程在问题一中已经详细介绍，在此不再赘述，我们使用随机森林预测化合物活性。模型预测效果在之后的章节给出。在此处使用的随机森林参数如下图所示：

5.4.2 极端梯度提升

极端梯度提升（XGBoost，eXtreme Gradient Boosting，XGboost）：极端梯度提升实际上是以集成学习为思想的，以多个分类回归树为基础分类器，采用梯度提升的方法进行训练，将多个预测器组合进行组合成一个强大的集成学习器，最后用以提升预测效果 [11]。

XGBoost 模型将基学习器称之为 CART，针对每一个 CART，其复杂度由 q 和叶子结点的输出 ω 决定。在本题中一个分子描述符 x，对应唯一的输出 ω。对于拥有 n 个样本，m 维特征的数据集 $D=\{(x_i,y_i)\}\,(|D|=n,x_i\in\mathbb{R}^m,y_i\in\mathbb{R})$ K 个 CART 预测的最终输出为：

$$\widehat{y_i}=\sum_{k=1}^{K}f_k\left(x_i\right),f_k\in F \quad (22)$$

其中 $F=\left\{f(x)=\omega_{q(x)}\right\}\left(q:\mathbb{R}^m\to T,\omega\in\mathbb{R}^T\right)$ 为 CART 构成的的集合；ω 为权重；T 为叶子节点个数；q 为表示每棵树的结构的向量, 由样本指向相应的叶子标签; 每个函数 f_k 对应一棵独立的树结构 qk 和叶子权重 k 。每棵 CART 的每个叶子节点对应一个连续分数值, i 代表第 i 个结点的分数。q(x) 是对样本 x 的打分, 即模型预测值。对于每个样本, 各个 CART 依据不同分类规则将它分类到叶子节点中, 通过累加对应叶子的分数 ω 来获得最终的预测结果。

5.4.3 梯度提升树

梯度提升树 (Gradient Boosting Decision Tree，GDBT)：是一种 boosting 算法，该算法由多棵 CART 回归树组成，所有树的结论累加起来做最终求解。换言之是在 GDBT 是在拟合上一个模型产生的残差，之后将多棵树的预测结果相加，从而最终决策。

5.4.4 集成学习

集成学习（Ensemble Learning）是一种能在各种的机器学习任务上提高准确率的强有力技术，其通过组合多个基学习器来完成学习任务。基学习器一般采用的是弱可学习器，通过集成学习，组合成一个强可学习器，一般情况来说，集成学习的方法会比单独基学习器的效果更好。本文采用的集成学习框架如图所示：

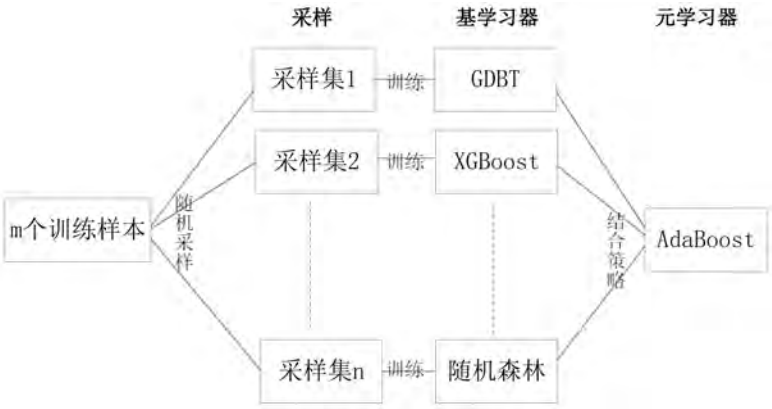


图 5-11　集成学习框架

本文采用 GDBT、XGBoost、随机森林作为基学习器，采用 AdaBoost 作为元学习器。学习器之间的相关性如下图所示。

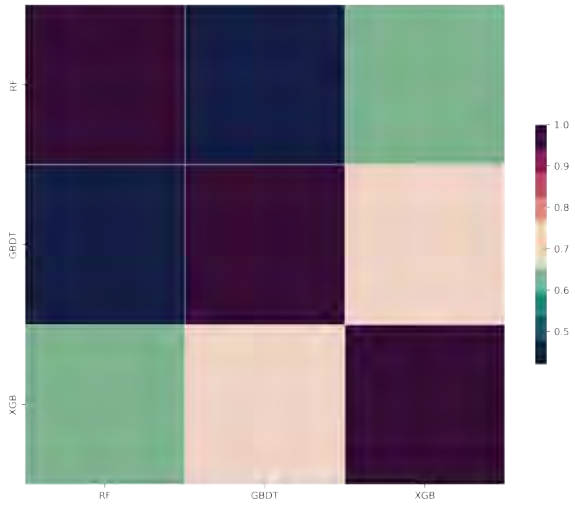


图 5-12　基模型的相关性

5.5 模型效果

我们在定性和定量两个角度展开模型分析。

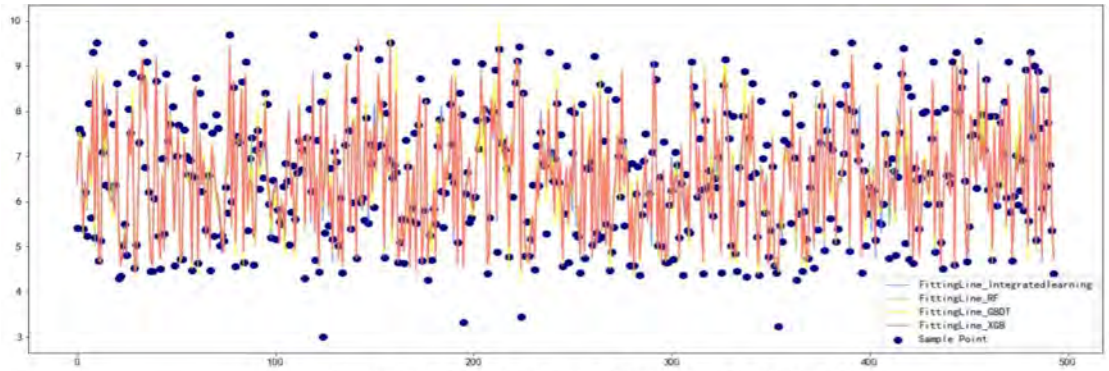
5.6 MIC+RFE+ 四类预测算法的效果

在该特征选择的前提下，我们进行预测模型建模所涉及的参数如下表:

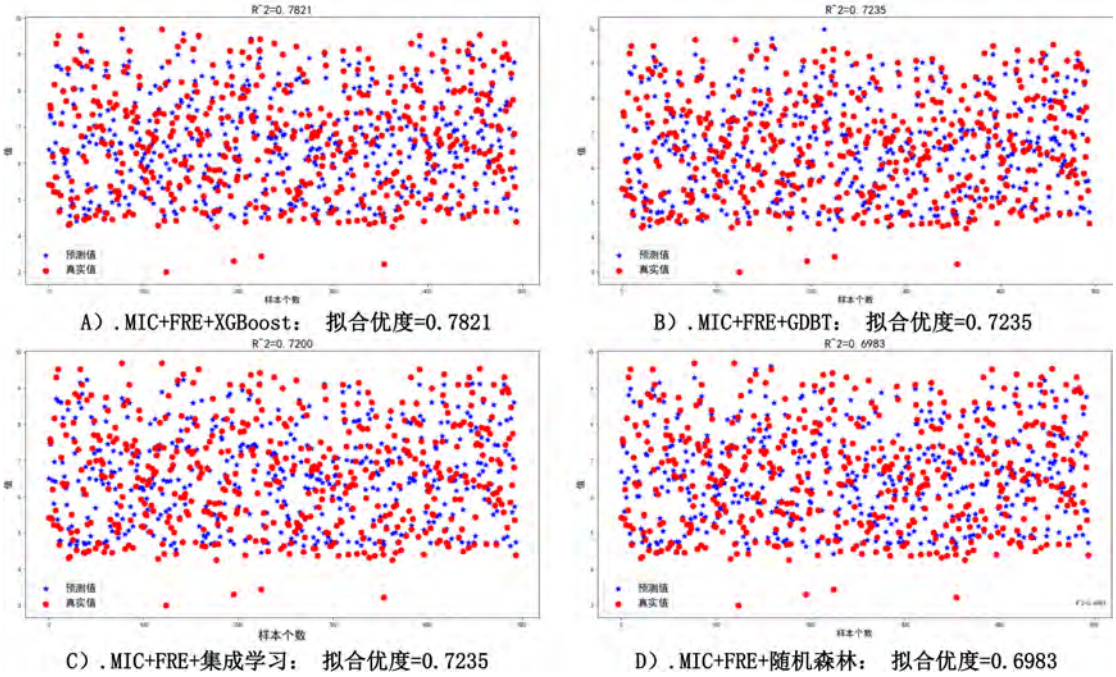
表 2　MIC+RFE+ 四类预测算法的模型参数

模型	参数设置
RF	n_estimators=108,max_depth =20
GBDT	n_estimators=295,max_depth=5,loss='ls',subsample=0.7
XGBOOST	{'silent':True,'obj':'reg:linear',"subsample":0.8,"max_depth":5,"eta":0.1,"gamma":0.02,"lambda":1,"alpha":0,"colsample_bytree":0.8,"colsample_bylevel":1,"colsample_bynode":1,"nfold":5}　num_round = 139
集成学习	XGBOOST {'learning_rate':0.1,'n_estimators':139,'max_depth':5,'min_child_weight':3,'gamma':0.02,'subsample':0.8,'colsample_bytree':0.7,'reg_alpha':1,'objective':'reg:linear'} RF　{'n_estimators': 108,'max_depth':20} GBDT{'n_estimators':295,'max_depth':5}

我们使用 MIC+FRE 进行特征选择，之后分别选择四类预测算法进行活性预测，结果如下所示。



MIC+FRE 进行特征选择+四种预测方法对活性预测的对比



A）.MIC+FRE+XGBoost：　拟合优度=0.7821

B）.MIC+FRE+GDBT：　拟合优度=0.7235

C）.MIC+FRE+集成学习：　拟合优度=0.7235

D）.MIC+FRE+随机森林：　拟合优度=0.6983

图 5-13　MIC+RFE+ 四类预测算法的的效果

我们发现，这几种算法中 XGboost 取得了最好的结果，通过分析其原因，总结如下。本题数据中存在稀疏性，XGboost 考虑了训练数据为稀疏值的情况，可

以为缺失值或者指定的值指定分支的默认方向，这能大大提升算法的效率。

5.7 Dco+GMM+ 四类预测算法的效果

在该特征选择的前提下，我们进行预测模型建模所涉及的参数如下表：

表 3　Dco+GMM+ 四类预测算法的参数

模型	参数设置
RF	n_estimators=243,max_depth =23
GBDT	n_estimators=294,max_depth=4,loss='ls',subsample=0.7
XGBOOST	{'silent':True,'obj':'reg:linear',"subsample":0.7,"max_depth":7,"eta":0.1,"gamma":0.03,"lambda":1,"alpha":0,"colsample_bytree":0.8,"colsample_bylevel":1,"colsample_bynode":1,"nfold":5}　num_round = 91
集成学习	XGBOOST {'learning_rate':0.1,'n_estimators':139,'max_depth':5,'min_child_weight':3,'gamma':0.02,'subsample':0.8,'colsample_bytree':0.7,'reg_alpha':1,'objective':'reg:linear'} RF　{'n_estimators': 243,'max_depth':23} GBDT{'n_estimators':295,'max_depth':5}

我们使用 Dco+GMM 进行特征选择，之后分别选择四类预测算法进行活性预测，结果如下所示。

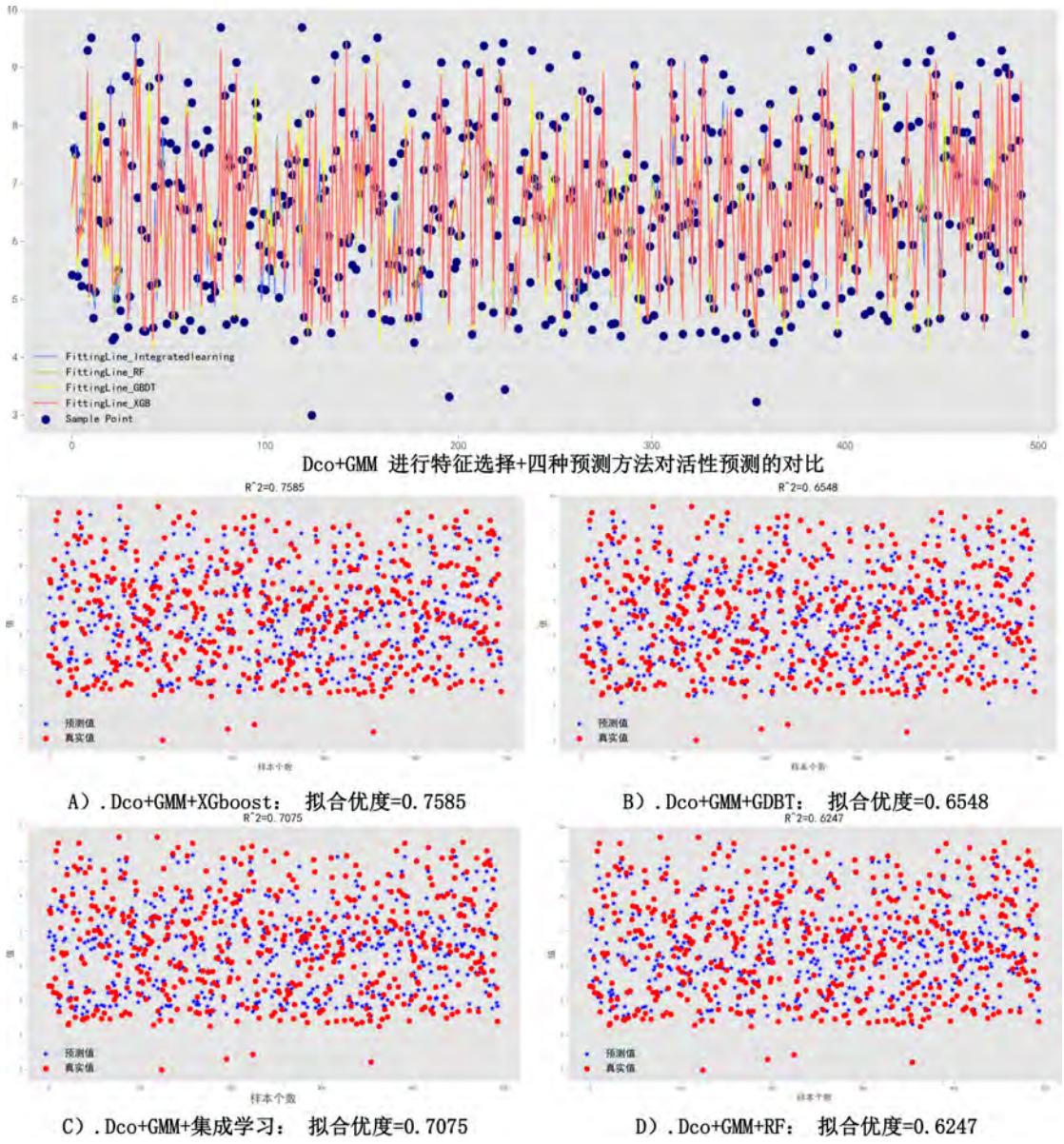


图 5-14　Dco+GMM+ 四类预测算法的的效果

与上一节相似的，我们得到了相似的结论，XGboost 效果最好，拟合优度 R^2 得分最高。

5.8 改进的随机森林评分法 + 四类预测算法的效果

在该特征选择的前提下，我们进行预测模型建模所涉及的参数如下表:

表 4　改进随机森林 + 四类预测模型的参数

模型	参数设置
RF	n_estimators=243,max_depth =29,random_state=420
GBDT	n_estimators=292,max_depth=5,loss='ls',subsample=0.7
XGBOOST	{'silent':True,'obj':'reg:linear',"subsample":0.6,"max_depth":4,"eta":0.1,"gamma":0.03,"lambda":1,"alpha":0,"colsample_bytree":0.8,"colsample_bylevel":1,"colsample_bynode":1,"nfold":5}　num_round = 200
集成学习	xgboost 基模型 {'learning_rate':0.1,'n_estimators':200,'max_depth':4,'min_child_weight':3,'gamma':0.03,'subsample':0.6,'colsample_bytree':0.7,'reg_alpha':1,'objective':'reg:linear'} RF 基模型　{'n_estimators': 243,'max_depth':29} XGBT　{'n_estimators':292,'max_depth':5,'loss':'ls'}

我们使用改进的随机森林评分法进行特征选择，之后分别选择四类预测算法进行活性预测，结果如下所示。

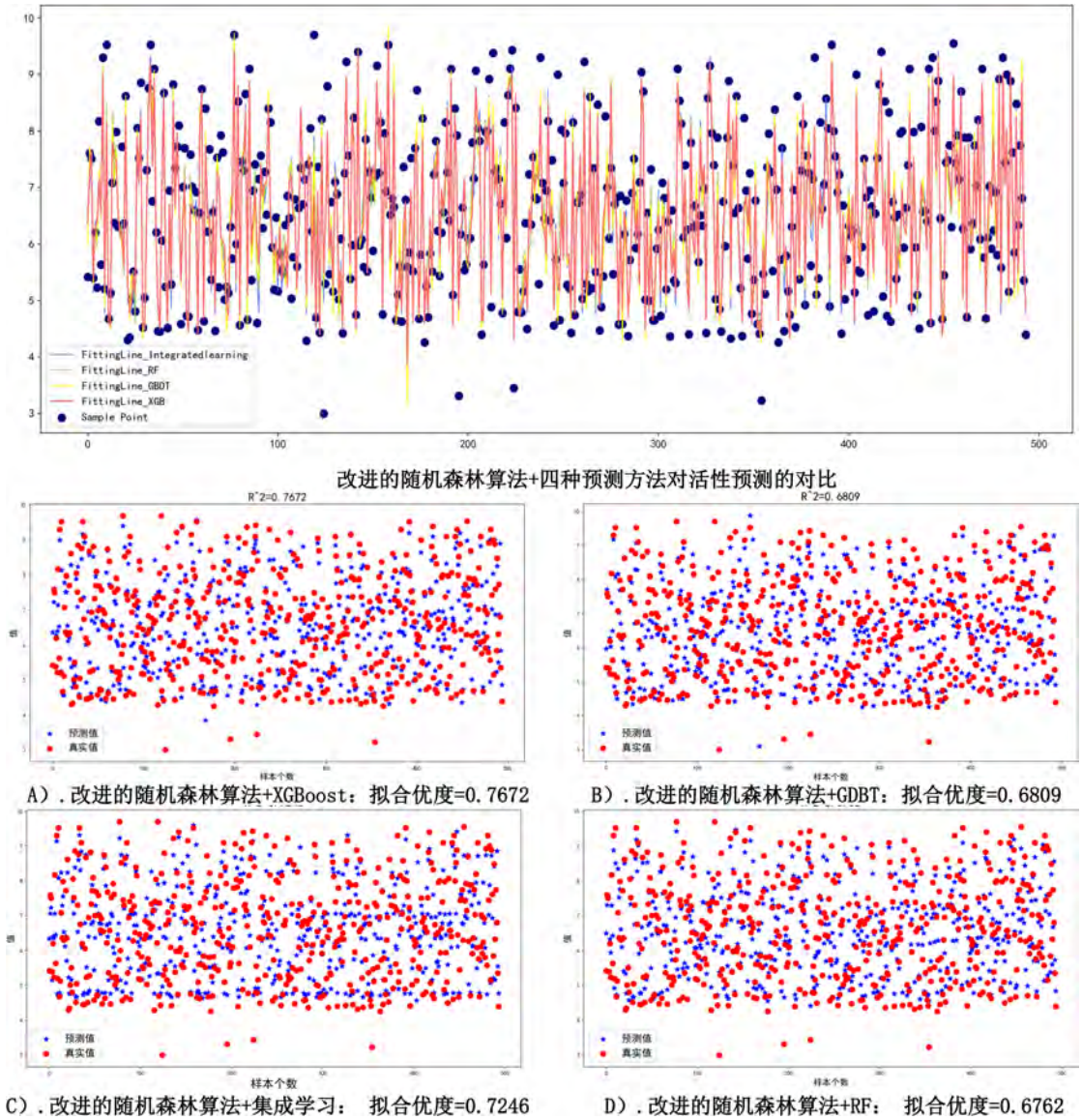


图 5-15　Dco+GMM+ 四类预测算法的的效果

与上一节相似的，XGboost 效果最好，由于数据存在稀疏性，其他模型难以

解决，所以我们认为在本任务中，XGBoost 进行生物活性预测可以取得最优解。

本方案的定量效果展示如下表所示。

表 5　对比实验结果

	RF	XGBoost	GDBT	集成学习
Dco+GMM	0.6247	0.7585	0.6548	0.7075
MIC+FRE	**0.6983**	**0.7821**	**0.7235**	0.7235
改进随机森林算法	0.6762	0.7672	0.6809	0.7246

我们发现使用 MIC+FRE 进行特征提取，所取得模型预测的结果拟合优度 R^2 值最高，这也说明这种策略进行特征选择最为合理，特征名称如下表。同时，我们发现 XGBoost 更适用于本题中化合物生物活性的预测，可以很好的解决数据稀疏性的问题。

表 6　最优分子描述符

序号	分子描述符
1	MDEC-23
2	LipoaffinityIndex
3	C1SP2
4	minsOH
5	minHsOH
6	maxHsOH
7	CrippenLogP
8	MLFER_A
9	BCUTc-11
10	VPC-6
11	BCUTc-1h
12	SHBint10
13	MDEC-33
14	SPC-6
15	maxssO
16	mindssC
17	VC-5
18	MAXDP
19	minHBa
20	TopoPSA

5.9 特征独立性检验

在此处为了检验我们提取分子描述符方法的性能，我们进行了特征独立性检验。我们分别计算这三种方法提取的分子描述符的斯皮尔曼相关系数，计算结果如下图所示。

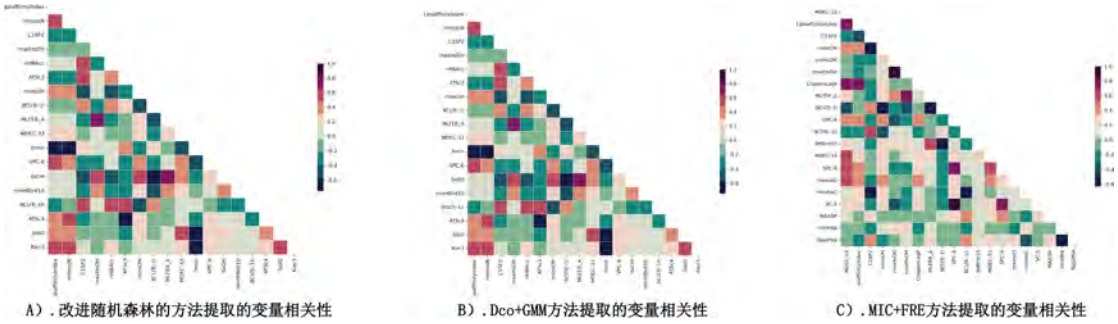


图 5-16　三种特征提取结果的独立性检验

我们可以发现，三种方法所提取的分子描述符之间无显著相关关系，因此通过独立性检验。

6. 问题三：模型的建立与求解

6.1 问题分析

问题三要求我们针对文件“ADMET.xlsx”中提供的 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。针对于本题目，我们首先对数据进行降维处理，之后对化合物的五种 ADMET 性质建立预测模型。

本问的解题思路如下图：

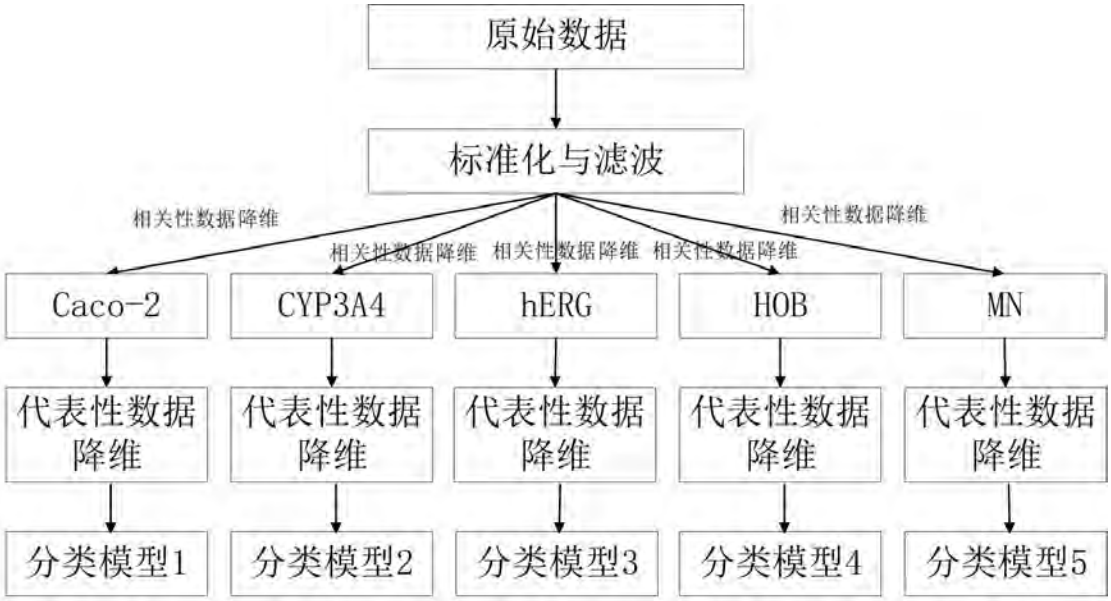


图 6-1　本问处理流程

6.2 数据降维

在数据降维前，我们对数据进行归一化处理，并且按照 4:1 划分训练集和测试集，并使用 K 折交叉验证。因为数据维度庞大，难以通过一次降维选择出构建分类模型的因变量。因此在本问中，我们分别在数据相关性角度和数据独立性角度，进行数据降维。其中使用随机森林打分方法对相关性进行降维，分别使用遗传算法对数据进行降维，过滤强耦连变量。使用随机森林方法进行相关性分析在之前过程中已经介绍过，在此不再赘述。考虑到题目给出的分子描述符之间存在着明显的非线性与偶联关系，因此仅使用皮尔逊相关系数等非线性评分方法，来选出具有代表性的分子描述符是不准确的。因为相关系数只考虑了单个 ADMET 性质与分子描述符之间的关系，而分子描述符之间、分子描述符的组合对 ADMET 性质的影响都没有考虑；然而遗传算法是从问题解的串集开始搜索，并且可以同时处理群体中的多个个体，从而减少了陷入局部最优解的风险。综合上述原因，我们选择遗传算法对变量进行二次降维。

遗传算法的运算流程如下所示。

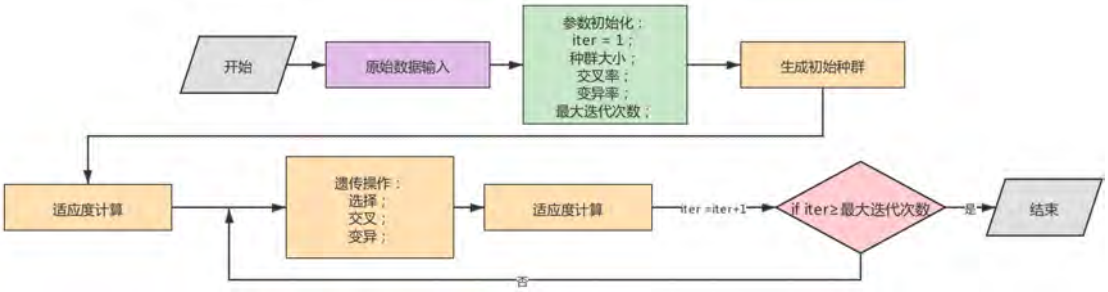


图 6-2　遗传算法流程

使用遗传算法的关键是，设置合理的适应度函数，本文中我们将 ADMET 性质的分类的精度作为目标函数，通过分析不同分子描述符对 ADMET 性质的影响，从而选择出用于分类的特征。这就达到了数据降维的目标。

本文中遗传算法的参数设置如下表所示。

表 7　本问中遗传算法的参数设置

模型	参数设置

GeneticSelectionCV	estimator, #模型
	cv=5, #交叉验证
	verbose=0
	scoring="accuracy",#评价标准
	max_features=25,#最多模型特征
	n_population=100, #种群大小
	crossover_proba=0.5,#交叉概率
	mutation_proba=0.2, #变异概率
	n_generations=150,#迭代次数

6.2.1 关于分类 Caco-2 任务的数据降维

本问中我们首先使用随机森林分析分子描述符与 Caco-2 性质的相关性，并保留排名前 60 的分子描述符，其结果如下所示。

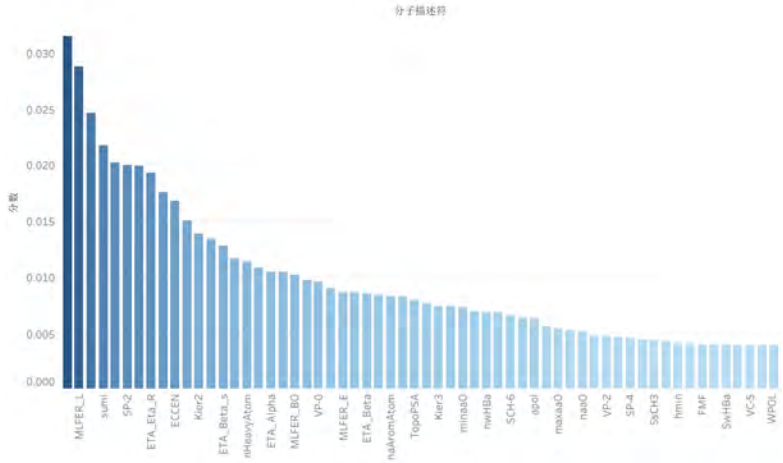


图 6-3　与 Caco-2 性质相关的前 60 的分子描述符

之后我们使用遗传算法，选取对分类模型具有代表性的分子描述符共计 11 个，如下表所示。

表 8　用于 Caco-2 性质分类的描述符

序号	分子描述符	序号	分子描述符
1	MLFER_L	12	TopoPSA
2	ETA_Eta_R_L	13	minaaO
3	sumI	14	SCH-6
4	ETA_Eta_F_L	15	maxaaO
5	WPATH	16	SsCH3
6	ECCEN	17	hmin
7	ETA_Eta_F	18	SP-6
8	ETA_Beta_s	19	FMF
9	MLFER_S	20	MDEC-12
10	SP-0	21	VC-5
11	MLFER_E	22	minwHBa

6.2.2 关于分类 CYP3A4 任务的数据降维

本问中我们首先使用随机森林分析分子描述符与 CYP3A4 性质的相关性，并保留排名前 60 的分子描述符，其结果如下所示。

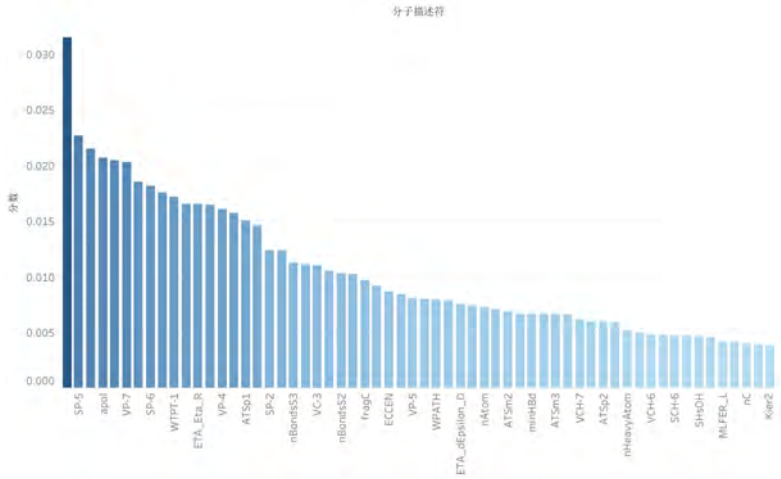


图 6-4　与 CYP3A4 性质相关的前 60 的分子描述符

之后我们使用遗传算法，选取对分类模型具有代表性的分子描述符共计 14 个，如下表所示。

表 9　用于 CYP3A4 性质分类的描述符

序号	分子描述符
1	SP-4
2	VP-7
3	VP-2
4	WTPT-1
5	VP-3
6	SP-3
7	SP-7
8	ETA_Eta
9	ETA_Beta_s
10	bpol
11	ETA_dEpsilon_D
12	ETA_Eta_L
13	SCH-6
14	SCH-7

6.2.3 关于分类 hERG 任务的数据降维

本问中我们首先使用随机森林分析分子描述符与 hERG 性质的相关性，并保留排名前 60 的分子描述符，其结果如下所示。

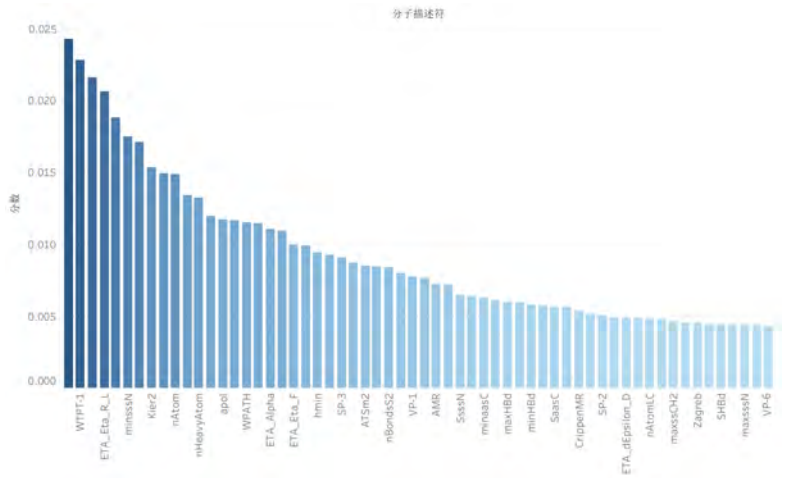


图 6-5　与 hERG 性质相关的前 60 的分子描述符

之后我们使用遗传算法，选取对分类模型具有代表性的分子描述符共计 7 个，如下表所示。

表 10　用于 hERG 性质分类的描述符

序号	分子描述符
1	VP-0
2	McGowan_Volume
3	ECCEN
4	bpol
5	WPATH
6	minaasC
7	maxaaCH

6.2.4 关于分类 HOB 任务的数据降维

本问中我们首先使用随机森林分析分子描述符与 HOB 性质的相关性，并保留排名前 60 的分子描述符，其结果如下所示。

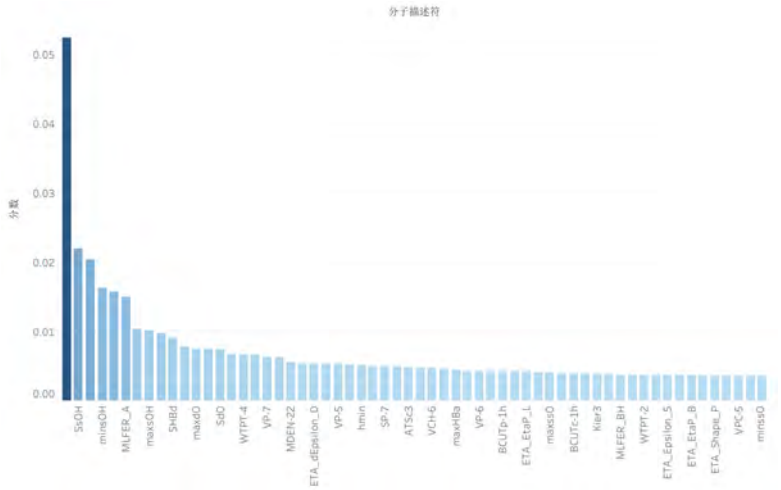


图 6-6　与 HOB 性质相关的前 60 的分子描述符

之后我们使用遗传算法，选取对分类模型具有代表性的分子描述符共计 7 个，如下表所示。

表 11　用于 HOB 性质分类的描述符

序号	分子描述符
1	BCUTc-11
2	SHsOH
3	maxsOH
4	SdO
5	VP-3
6	VP-6
7	SPC-6

6.2.5 关于分类 MN 任务的数据降维

本问中我们首先使用随机森林分析分子描述符与 MN 性质的相关性，并保留排名前 60 的分子描述符，其结果如下所示。

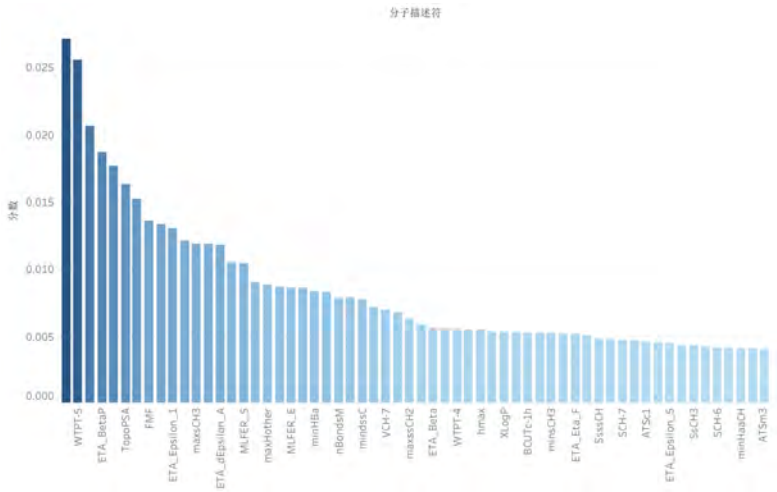


图 6-7　与 MN 性质相关的前 60 的分子描述符

之后我们使用遗传算法，选取对分类模型具有代表性的分子描述符共计 11 个，如下表所示。

表 12　用于 MN 性质分类的描述符

序号	分子描述符
1	maxssCH2
2	WTPT-5
3	ETA_BetaP_s
4	nN
5	ATSc2
6	ETA_EtaP_L
7	BCUTc-1h
8	SHCsats
9	ETA_Epsilon_5
10	ETA_EtaP_B
11	maxsCH3

6.3 分类模型建立

我们需要针对上述选出的具有独立性和代表性分子描述符，建立其对应的 ADMET 五种性质的分类模型。选择模型对数据进行分类，最首要的就是对数据进行分析，从而决定如何选择模型。对此我们首先分析了分子描述符之间，以及分子描述符与这五种性质的关系。我们以 Seaborn 图绘制，如下所示：

图 6-8　针对 Caco-2 的分类变量关系图

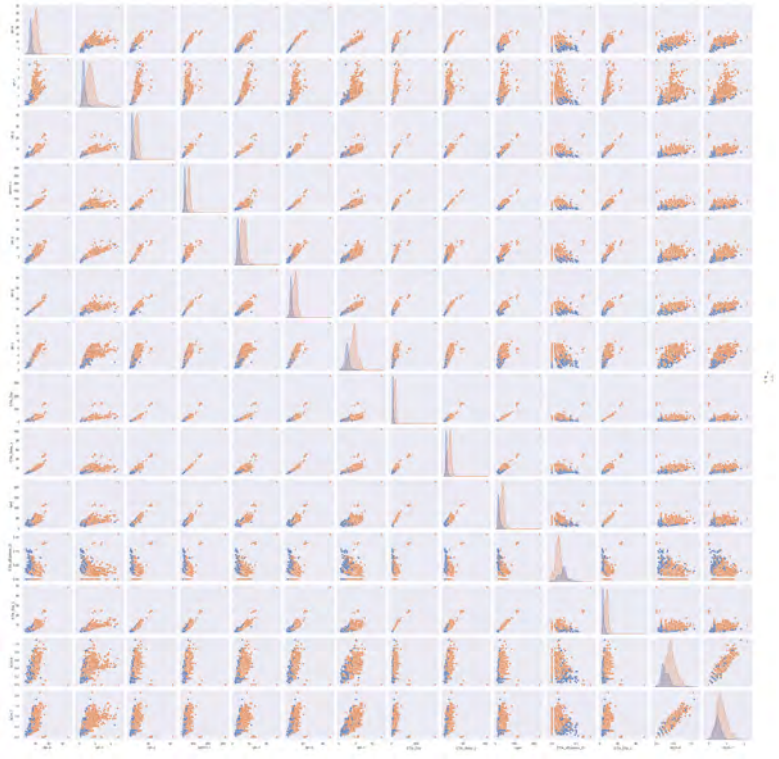


图 6-9　针对 CYP3A4 的分类变量关系图

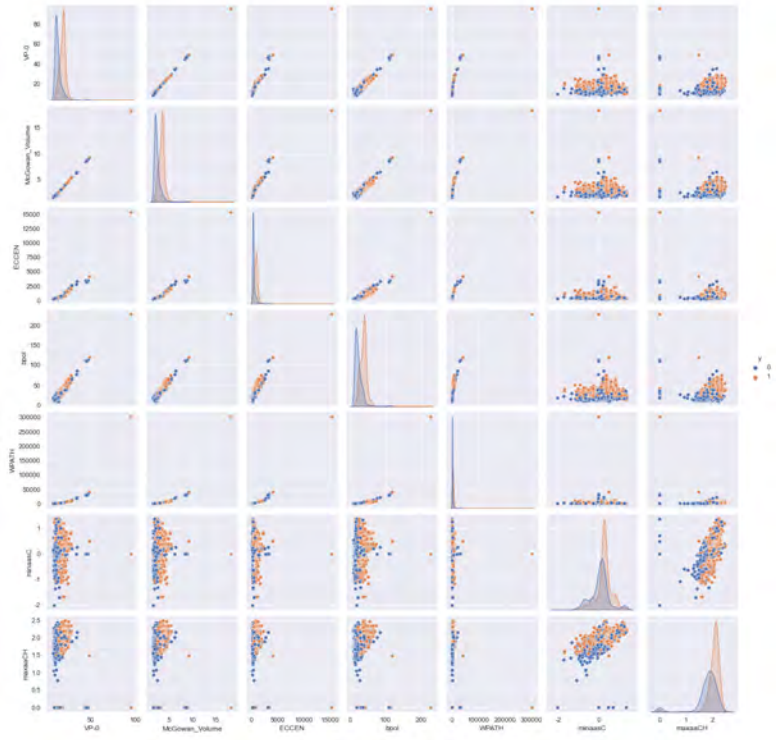


图 6-10　针对 hERG 的分类变量关系图

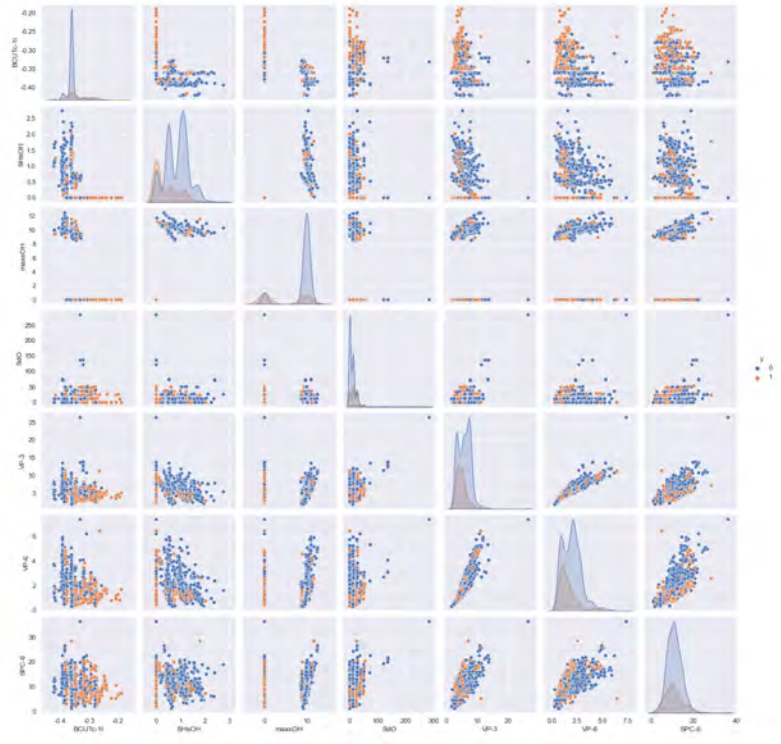


图 6-11　针对 HOB 的分类变量关系图

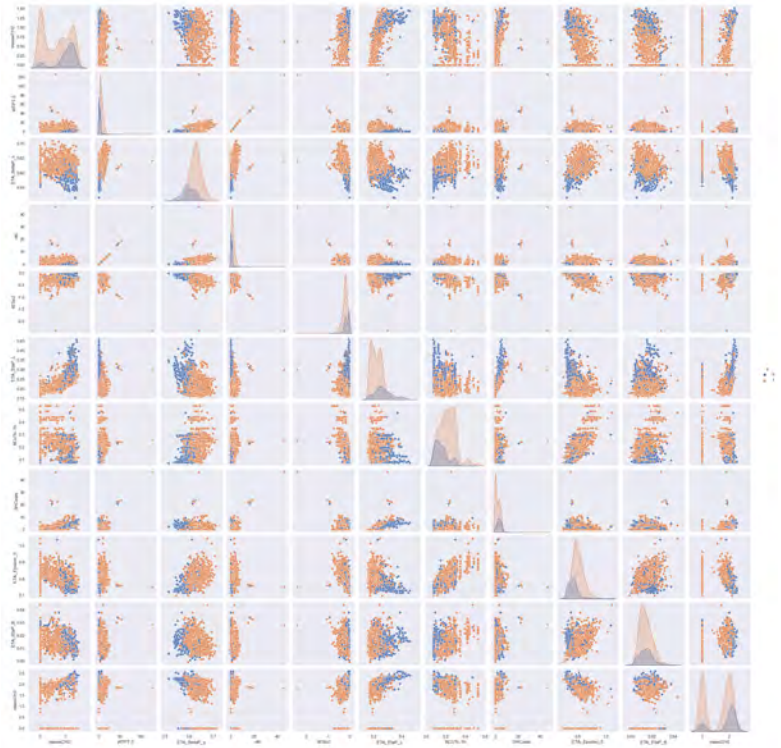


图 6-12　针对 MN 的分类变量关系图

我们在上面的五张图中，可以发现 ADMET 五种性质与分子描述符之间存在着明显的非线性，以及数据稀疏性并且分子描述符的独立性较强。此类数据分

布特性，首先我们不宜选取线性模型进行分类，其次我们不宜选择深度模型，因为数据稀疏性以及数据样本较少，所以会导致过拟合，综合上述特征，对这五种特性进行分类，我们选择使用 XGBoost，实验结果证明该方法适用于此类数据，并且会取得较好的拟合效果。并且为了验证树模型对此类问题的效能，我们还将随机森林设置为一组对比方法，定性的实验结果如下表所示。

表 13　用于 MN 性质分类的描述符

	Random Forest	xgboost
Caco-2	0.900809717	0.912955466
CYP3A4	0.925101215	0.92902834
hERG	0.852226721	0.882591093
HOB	0.858299595	0.868421053
MN	0.937246964	0.963562753

五种性质的 ROC 曲线如下图所示。

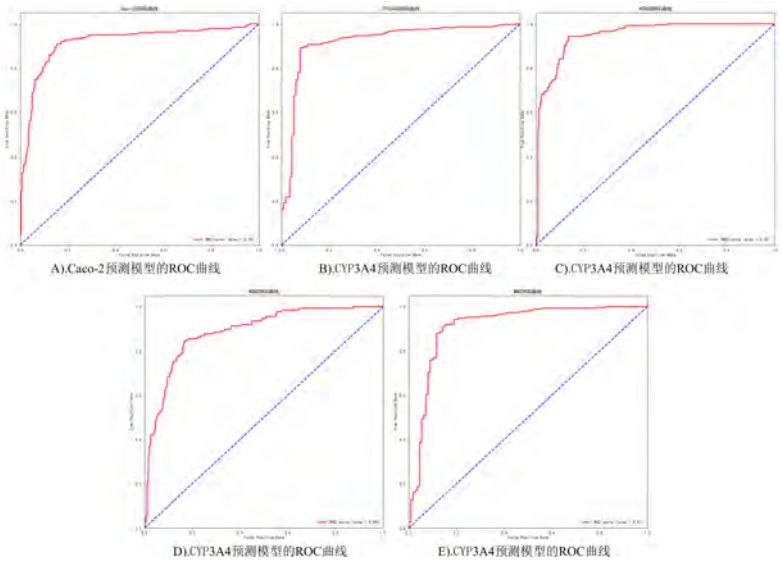


图 6-13　五种基于 XGBoost 分类模型的 ROC 曲线

我们发现与随机森林相比，XGboost 模型效果更好，原因是 XGboost 集成了随机森林的优点，并在此上进行了算法优化。学习模型更具有鲁棒性，便于应用在多类数据集中 [12]。

7. 问题四：模型的建立与求解

7.1 问题分析

本问旨在寻找分子描述符，以及分子描述符取值或范围，能够使化合物对抑制 ERα 具有更好的生物活性，并且具有更好的 ADMET 性质。针对于此，我们将其视为多目标优化问题，基于第二、三问构建的预测模型，以生物活性最高和 ADMET 性质最好作为目标，以二、三两问筛选出的分子描述符作为决策变量，构建优化模型。考虑到本优化问题搜索空间大，全局最优解寻找难度大的特点，我们采用了遗传算法对这一优化问题进行求解。由于分子描述符超过一定范围在应用中不具有实际意义，我们利用所给数据的范围来表示分子描述符的取值范围。利用 Python 编程实现，得到了一组分子描述符和其相应的取值，使得化合物对抑制 ERα 具有更好的生物活性，同时具有更好的 ADMET 性质。最后，进行算法对比分析，证明了遗传算法的有效性和优越性。我们解决本文的整体流程如下所示。

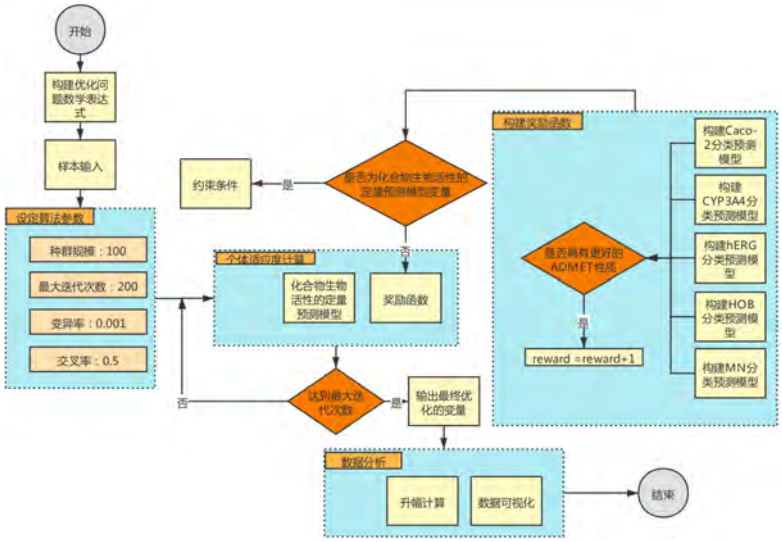


图 7-1　解决问题四的流程

本问思维导图如下图所示:

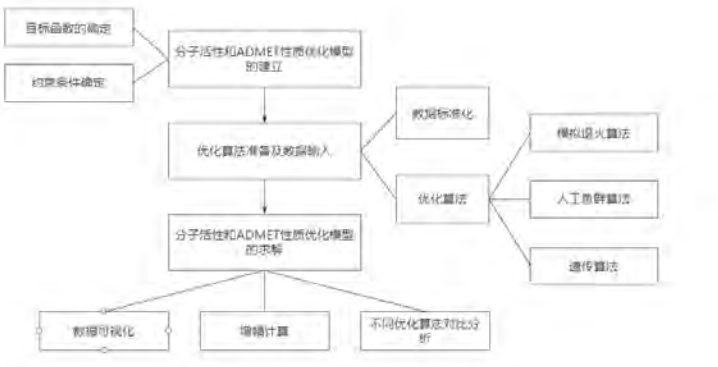


图 7-2　本问思维导图

7.2 模型介绍

我们使用遗传算法进行优化，并且使用模拟退火算法和人工鱼群算法作为对照组，以证明本文方案的优势。

7.2.1 模拟退火算法

模拟退火算法 (Simulated Annealing，SA) 是基于 Monte-Carlo 迭代求解策略的一种智能算法。模拟退火算法从某一较高的初始温度开始，随着温度不断下降，结合概率突跳特点在解空间中寻找全局最优解，避免陷入局部最优。模拟退火算法在理论上具有概率的全局最优性能 [15]。

7.2.2 人工鱼群算法

人工鱼群算法是一种群体智能算法。自然界中，鱼往往可以自行或者尾随其他的鱼找到营养物质最多的地方进行觅食，人工鱼群算法利用这一特点，模仿鱼群的觅食、聚群、追尾等过程，实现智能寻优过程 [14]。

(1) 觅食行为：一般情况下鱼在水中随机地自由游动, 当发现食物时, 则会向食物逐渐增多的方向快速游去。

(2) 聚群行为: 鱼在游动过程中为了保证自身的生存和躲避危害会自然地聚集成群，鱼聚群时所遵守的规则有三条：分隔规则：尽量避免与临近伙伴过于拥挤; 对准规则: 尽量与临近伙伴的平均方向一致; 内聚规则: 尽量朝临近伙伴的中心移动。

(3) 追尾行为：当鱼群中的一条或几条鱼发现食物时, 其临近的伙伴会尾随其快速到达食物点。

(4) 随机行为：单独的鱼在水中通常都是随机游动的，这是为了更大范围地寻找食物点或身边的伙伴。

7.2.3 遗传算法

遗传算法（Genetic Algorithm, GA）是模拟遗传学机理的生物进化过程的计算模型，是一种智能算法。其主要优点就是具有很好的全局寻优能力，并且遗传算法采用概率化的寻优方法，可以自适应地调整搜索方向。遗传算法中的遗传操作有交叉、选择以及变异 [13]。常见的遗传算法流程如下：

（1）初始化种群规模 $size_pop$，用随机数产生染色体每个基因的值，这些值符合其约束范围。初始化当前进化代数 iter。

（2）计算种群中的全部染色体的适应度，保存适应度最大的染色体 Best。

（3）采用轮盘赌方式对种群的染色体进行选择操作。

（4）根据交叉概率 $prob_cro$ 选择父代染色体进行交换部分基因操作，产生新的子代染色体取代父代染色体。

（5）新种群中的染色体的基因会根据变异概率 $prob_mut$ 发生变异。

（6）进化代数 iter 加 1。如果 iter 超过最大进化代数，算法结束，否则返回（3）。

7.3 优化模型建立

7.3.1 优化目标及约束设定

(1) 决策变量：本文所建立的模型中影响化合物生物活性以及 ADMET 性质的分子描述符一共有 70 个，其中影响化合物生物活性的有 20 个，影响 ADMET 性质的有 55，其中存在 5 个重复变量。决策变量记为：

$$X = \{x_1, x_2, ..., x_{70}\} \quad (23)$$

(2) 目标函数：

$$Max[F(x), Reward(g_i(x))] \quad (24)$$

其中 $F(x)$ 为化合物生物活性的预测函数，$g_i(x), i = 1, 2..., 5$ 分别代表 ADMET5 个性质的分类模型。$Reward(g_i)$ 为奖励函数，其代表化合物 ADMET 五种较好性质的求和。$Reward(g_i) = g_1 + g_2 + (1 - g_3) + g_4 + (1 - g_5)$ 其中 g_1 代表 Caco-2 的分类模型，输出为 0 和 1，‘1’代表该化合物的小肠上皮细胞渗透性较好，‘0’代表该化合物的小肠上皮细胞渗透性较差。g_2 代表分类 CYP3A4 的模型，输出为 0 和 1。其中，‘1’代表该化合物能够被 CYP3A4 代谢，‘0’代表该化合物不能被 CYP3A4 代谢。g_3 代表 hERG 的分类模型，输出为 0 和 1。其中：‘1’代表该化合物具有心脏毒性，‘0’代表该化合物不具有心脏毒性。g_4 代表 HOB 的分类模型，其输出为 0 和 1，‘1’代表该化合物的口服生物利用度较好，‘0’代表该化合物的口服生物利用度较差。g_5 代表是 MN 的分类模型，其输出为 0 和 1。‘1’代表该化合物具有遗传毒性，‘0’代表该化合物不具有遗传毒性。在本文中，我们认为 Caco-2 取 1，CYP3A4 取 1，hERG 取 0，HOB 取 1，MN 取 0 为最优组合。此时奖励函数 $Reward = 5$。

(3) 约束条件：

我们设计的约束条件如下式

$$s.t = \begin{cases} x_{min} \leqslant x_i \leq x_{max} \\ 3 \leq reward(g_i) \leq 5 \end{cases}$$

其中存在四个分子描述符为离散型变量，其余均为连续型变量。我们对离散型变量采取整数约束。

(4) 模型参数设定：

本文使用遗传算法进行优化，并且采用人工鱼群算法和模拟退火算法作为对比。这三种模型参数的设定如下表所示。

表 14　优化算法参数设定

优化算法	参数
遗传算法	种群规模 size_pop = 100 最大进化代数 max_iter = 200 变异概率 prob_mut=0.001 交叉概率 cor_mut = 0.5
人工鱼群算法	种群规模 size_pop = 50 最大迭代次数 max_iter = 50 最大尝试捕食次数 max_try_num = 50 每一步的最大位移比例 step = 10 鱼的最大感知范围 visual = 20 鱼的感知范围衰减系数 q = 0.98 拥挤度阈值 delta = 1
模拟退火算法	最大温度 T_max = 1 最小温度 T_min = 0.0000000001 链长 L = 300 冷却耗时 max_stay_counter = 150

(5) 模型求解：

我们同时使用三种优化算法对其进行求解，优化效果如下所示。

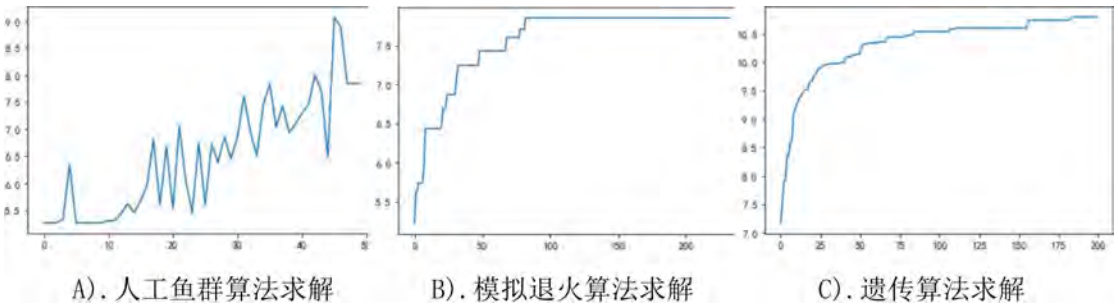


图 7-3　优化效果展示

可以发现遗传算法的效果最优，可以发现人工鱼群算法在此问题中效果不佳，由于群体算法随机求解的本质，因此面对高度稀疏性数据难以收敛。

我们通过遗传算法求解，找到如下分子描述符，以及分子描述符在如下取值时，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，优化增幅可达 4.3%。

表 15　优化后的分子描述符最佳取值

分子描述符	取值	分子描述符	取值	分子描述符	取值
ATSc2	-0.938542799	SP-0	107.600548	minaaO	2.23512618
BCUTc-1h	0.237889565	SP-3	43.8208144	minaasC	0.373581646
BCUTc-1l	-0.273273568	SP-4	33.5311887	minwHBa	1.75636472
ECCEN	4988	SP-6	17.1110795	nN	10
ETA_BetaP_s	0.575998903	SP-7	6.00690599	sumI	221.27007
ETA_Beta_s	62.8200538	SPC-6	34.0161819	MDEC-23	49.96407764
ETA_Epsilon_5	0.923601128	SdO	40.3201311	LipoaffinityIndex	13.28254605
ETA_Eta	279.392411	SsCH3	18.4719364	C1SP2	20
ETA_EtaP_B	0.005004499	TopoPSA	31.2414525	minsOH	8.99901983
ETA_EtaP_L	0.439303115	VC-5	0.117741904	minHsOH	0.70352262
ETA_Eta_F	14.4959704	VP-0	18.9188393	maxHsOH	0.63304089
ETA_Eta_F_L	9.47490125	VP-2	35.8179608	CrippenLogP	5.25225823
ETA_Eta_L	25.8029171	VP-3	16.7382702	MLFER_A	1.1149264
ETA_Eta_R_L	23.0644649	VP-6	6.44174965	VPC-6	11.60093146
ETA_dEpsilon_D	0.081170272	VP-7	2.02762625	SHBint10	9.84926562
FMF	0.018028727	WPATH	301690	MDEC-33	14.78321616
MDEC-12	0.982611106	WTPT-1	205.368576	maxssO	6.64048951
MLFER_E	2.01562242	WTPT-5	40.3989124	mindssC	-1.48911605
MLFER_L	12.5056998	bpol	23.8902496	MAXDP	7.56602459
MLFER_S	8.45392276	hmin	-0.550539385	minHBa	-0.99791122
McGowan_Volume	9.32188168	maxaaCH	1.42137784	pIC50	10.790529
SCH-6	0.297401709	maxaaO	3.70338114	较优 ADMET 性质个数	5
SCH-7	0.77069671	maxsCH3	0.312007362		
SHCsats	42.6408752	maxsOH	8.8157027		
SHsOH	1.28965408	maxssCH2	0.777172116		

8. 问题总结

第一问通过数据分析，我们发现自变量和因变量间存在着较强的非线性关系，皮尔逊相关系数并不能很好地衡量这一情形，故选择采用最大信息系数或距离相关系数来进行相关性的衡量。其次，数据样本的维度极大，采用单一降维的方式可能不能很好地考虑样本数据中可能存在的诸多问题，故采用三次逐步降维的方式，即灰色关联度 + 距离相关系数 + 随机森林单变量打分来选出和因变量最为密切的 20 个特征。

第二问需要建立 ERα 生物活性关于分子描述符的预测模型，难点主要有二：应该采用何种的特征选择和预测模型，如何验证此种模型的优越性。首先，结合本问题的数据特征，我们初步选定了最大信息系数（MIC）加递归特征消除（RFE），距离相关系数 + 高斯混合模型（GMM），改进的随机森林打分模型等三种方案。其次，对于每种特征选择方案，我们分别建立了随机森林，梯度提升，集成学习，极度梯度提升四种预测模型，并给出了可视化的结果。最后，我们通过对比实验分析验证法，分析了不同模型的优劣，最终发现基于递归特征消除（RFE）和极度梯度提升（XGBoost）模型的效果最好。

问题三需要分别构建化合物的 Caco-2、CYP3A4、hERG、HOB、MN 的分类预测模型。对每个分类变量，我们分别使用随机森林进行打分，并取出排名前 60 的变量用于后续的处理。接下来，将参考模型的 r^2 设置为优化目标，采用遗传算法对剩余特征进行优化迭代，最终获取构建分类预测模型的较优特征。在预测阶段，我们分别建立了随机森林模型和极度梯度提升树模型，并给出了它们的可视化结果和比较。

在问题四的求解中，我们对分子描述符进行优化，分别采用了遗传算法、人工鱼群算法、模拟退火算法对优化模型进行求解。智能优化算法可以用于处理非常复杂的函数，并且无需函数的具体表达式，随着迭代次数的增加，算法总能找到约束条件下的最优值。遗传算法具有较强地全局搜索能力，相较于其他两种算法，虽然收敛速度较慢，但更易求得全局最优解。

9. 参考文献

[1] 师金, 梁迪, 李道娟, 王立群, 靳晶, 张亚琛, 贺宇彤. 全球女性乳腺癌流行情况研究 [J]. 中国肿瘤,2017,26(09):683-690.

[2] 何冰, 罗勇, 李秉轲, 薛英, 余洛汀, 邱小龙, 杨登贵. 基于分子描述符和机器学习方法预测和虚拟筛选乳腺癌靶向蛋白 HEC1 抑制剂 [J]. 物理化学学报,2015,31(09):1795-1802.

[3] 王浩臻. 以 TGF-βI 型受体为靶点从现有药物数据库中进行抗肺纤维化候选药物筛选的初步研究 [D]. 辽宁大学,2021.

[4] 王志中. 雌激素类衍生物对雌激素受体结合活性的 QSAR 和分子对接研究 [D]. 西北农林科技大学,2012.

[5] 李运. 机器学习算法在数据挖掘中的应用 [D]. 北京邮电大学,2015.

[6] 张紫嫣, 包永睿, 王帅, 李天娇, 刘金瑛, 孟宪生. 灰色关联度法研究红蓼抗肿瘤的药用部位研究 [J]. 世界科学技术-中医药现代化,2019,21(03):419-423.

[7] 张璐, 孔令臣, 陈黄岳. 基于距离相关系数的分层聚类法 [J]. 计算数学,2019,41(03):320-334.

[8] 刘凯. 随机森林自适应特征选择和参数优化算法研究 [D]. 长春工业大学,2018.

[9] 吴辰文, 梁靖涵, 王伟, 李长生. 基于递归特征消除方法的随机森林算法 [J]. 统计与决策,2017(21):60-63.

[10] 田杰, 韩冬, 胡秋霞, 马孝义. 基于 PCA 和高斯混合模型的小麦病害彩色图像分割 [J]. 农业机械学报,2014,45(07):267-271.

[11] 孟凡宇. 基于极端梯度提升算法的癌症诊断分类研究 [D]. 大连海事大学,2020.

[12] 韩中庚. 数学建模方法及其应用 [M]. 高等教育出版社, 2005.

[13] 高飞. MATLAB 智能算法超级学习手册 [M]. 人民邮电出版社:, 201405.498.

[14] 李晓磊, 钱积新. 人工鱼群算法: 自下而上的寻优模式 [A]. 中国系统工程学会过程系统工程专业委员会. 过程系统工程 2001 年会论文集 [C]. 中国系统工程学会过程系统工程专业委员会: 中国系统工程学会,2001:7.

[15] 卢宇婷, 林禹攸, 彭乔姿, 王颖喆. 模拟退火算法改进综述及参数探究 [J]. 大学数学,2015,31(06):96-103.

附录 A 程序代码

```
#距离相关系数筛选部分
import dcor
test_data=n1.iloc[:,1:]
Y_data=np.array(n1['y'].tolist())
score_list=[]
for col in test_data.columns:
  x=test_data[col]
  score=dcor.distance_correlation(x,Y_data)
  score_list.append(score)
score_list

#基于随机森林的单变量打分
from sklearn.model_selection import cross_val_score,
   ShuffleSplit
from sklearn.ensemble import RandomForestRegressor
import numpy as np

X = n2.iloc[:,1:]
Y = n2['y'].tolist()
names = n2.columns.tolist()[1:]

rf = RandomForestRegressor(n_estimators=20, max_depth=4)
scores = []
# 单独采用每个特征进行建模，并进行交叉验证
for i in range(X.shape[1]):
  score = cross_val_score(rf, X.iloc[:, i:i+1], Y,
     scoring="r2",cv=ShuffleSplit(len(X), 3, .3))
  scores.append((format(np.mean(score), '.3f'), names[i]))
print(sorted(scores, reverse=True))
name_list3=[]
score_list3=[]
for index in scores:
  name_list3.append(index[1])
  score_list3.append(index[0])
final_re=pd.DataFrame({'col':name_list3,'score':score_list3})

#GBDT部分
rfc_s_list=[]
list_nes=list(range(240,300))
for i in list_nes:
  clf =
     GradientBoostingRegressor(n_estimators=i,max_depth=3,loss='ls')
  rfc_s =
     cross_val_score(clf,x_train,y_train,cv=10,n_jobs=-1).mean()
  rfc_s_list.append(rfc_s)
print(max(rfc_s_list),
   list_nes[rfc_s_list.index(max(rfc_s_list))])
plt.plot(list_nes,rfc_s_list)
plt.show()
rfc_m_list=[]
list_nes=list(range(1,14))
```

```
for i in list_nes:
    clf =
        GradientBoostingRegressor(n_estimators=295,max_depth=i,loss='ls')
    rfc_s =
        cross_val_score(clf,x_train,y_train,cv=10,n_jobs=-1).mean()
    rfc_m_list.append(rfc_s)
print(max(rfc_m_list),
     list_nes[rfc_m_list.index(max(rfc_m_list))])
plt.plot(list_nes,rfc_m_list)
plt.show()
rfc_su_list=[]
list_nes=[0.6,0.7,0.8,0.9,1]
for i in list_nes:
    clf =
        GradientBoostingRegressor(n_estimators=295,max_depth=5,loss='ls',subsampl
    rfc_s =
        cross_val_score(clf,x_train,y_train,cv=10,n_jobs=-1).mean()
    rfc_su_list.append(rfc_s)
print(max(rfc_su_list),
     list_nes[rfc_su_list.index(max(rfc_su_list))])
plt.plot(list_nes,rfc_su_list)
plt.show()
clf=GradientBoostingRegressor(n_estimators=295,max_depth=5,loss='ls',subsample
clf.fit(x_train,y_train)
predict_x3=clf.predict(x_test)
print(r2_score(predict_x3,y_test))
import matplotlib.pyplot as plt
plt.figure(figsize=[15,8])
plt.rcParams['font.family'] = ['sans-serif']
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.scatter(range(0,len(predict_x3)),predict_x3,c='b',marker='*',s=80,label="预
plt.scatter(range(0,len(y_test) ),y_test,c =
    'r',s=80,marker='o',label="真实值")
plt.legend(['真实值','预测值'],loc = 1,fontsize = 6)
plt.xlabel('样本个数',fontsize = 15)
plt.ylabel('值',fontsize = 15)
plt.legend(fontsize='xx-large')
plt.title('R^2=0.7235',fontsize = 20)
# plt.text(480,3,'R^2=0.7235')
plt.savefig(r'C:\Users\Admin\Desktop\202\2021年D题\基于RFE
    GBDT对试验样例的预测结果.png')
plt.show()

#集成学习部分
from sklearn.model_selection import cross_val_score
#创建数据集
dataset = Dataset(x_train,y_train,x_test)
model_rf=Regressor(dataset=dataset,estimator=RandomForestRegressor,parameters=
    108,'max_depth':20},name='rf')
#model_lr=Regressor(dataset=dataset,estimator=LinearRegression,
    parameters={'normalize': True},name='lr')
#model_rfc=Regressor(dataset=dataset,estimator=SVR,parameters={'kernel':'rbf',
#model_dtr=Regressor(dataset=dataset,estimator=DTR,parameters={'max_depth':4},
model_xgb=Regressor(dataset=dataset,estimator=XGBRegressor,parameters={'learni
```

```
model_gbdt=Regressor(dataset=dataset,estimator=GradientBoostingRegressor,param
# Stack two models
# Returns new dataset with out-of-fold predictions
pipeline = ModelsPipeline(model_rf,model_xgb,model_gbdt)
stack_ds = pipeline.stack(k=10,seed=111)#用于stacking
# Train LinearRegression on stacked data (second stage)
stacker = Regressor(dataset=stack_ds,
    estimator=GradientBoostingRegressor)
results4 = stacker.predict()
# Validate results using 10 fold cross-validation
result = stacker.validate(k=10,scorer=r2_score)

#xgboost部分
param =
    {'silent':True,'obj':'reg:linear',"subsample":0.8,"max_depth":5,"eta":0.1,"
num_round = 139
model = xgb.train(param, dfull, num_round)
from sklearn.metrics import r2_score
dtest = xgb.DMatrix(x_test)
r2_score(y_test,model.predict(dtest))

#汇总综合图部分
import matplotlib.pyplot as plt
xx=range(0,len(y_test))
plt.figure(figsize=(20,8))
plt.scatter(xx,y_test,color="#000080",label="Sample
    Point",linewidth=3)
plt.plot(xx,results4,color="#4876FF",label="FittingLine_Integratedlearning",li
plt.plot(xx,y_hat2,color="#A2CD5A",label="FittingLine_RF",linewidth=1)
plt.plot(xx,predict_x3,color="#FFFF00",label="FittingLine_GBDT",linewidth=1)
plt.plot(xx,predict_y,color="#FF6A6A",label="FittingLine_XGB",linewidth=1.5)
plt.rcParams.update({'font.size': 12})
plt.legend(loc='upper left')
plt.legend()
plt.savefig(r'C:\Users\Admin\Desktop\202\2021年D题\基于RFE
    横向对比.png')
plt.show()

#分布图部分
sp=oridata.corr('spearman')
fig_path='./随机森林打分+xgboost拟合结果图.png'
fig=sns.pairplot(sp)
scatter_fig = fig.get_figure()
scatter_fig.savefig(fig_path, dpi = 400)

#热力图部分
import seaborn as sns
import palettable
import matplotlib as mpl
mpl.rcParams['axes.unicode_minus'] = False
plt.figure(figsize=(11, 9),dpi=100)
fig=sns.heatmap(data=sp,vmax=1,
       cmap=palettable.cmocean.diverging.Curl_10.mpl_colors,
#          annot=True,
```

```
#           fmt=".2f",
        annot_kws={'size':8,'weight':'normal',
            'color':'#253D24'},
        mask=np.triu(np.ones_like(sp,dtype=np.bool)),#显示对脚线下面部分图
        square=True,
        linewidths=.5,#每个方格外框显示，外框宽度设置
        cbar_kws={"shrink": .5}
)
scatter_fig = fig.get_figure()
#低方差滤波部分
Y_data=np.array(data_selected['pIC50'].tolist())
score_list=[]
for col in test_data.columns:
  x=test_data[col]
  score=dcor.distance_correlation(x,Y_data)
  score_list.append(score)
score_list

#t-sne可视化部分
# #二维空间
ts = TSNE(n_components=2, init= 'pca' , random_state=3)
reslut = ts.fit_transform(need_data_norm)
plt = plot_embedding(reslut, labels2, 't-SNE Embedding of
   digits' )
plt.xlabel('x')
plt.ylabel('y')
plt.title('t-sne聚类结果二维可视化')
plt.savefig('./t-sne聚类结果二维可视化.png')
plt.show()

# 2.绘制图片
plt.figure("三维空间的映射",figsize=[8,8], facecolor="lightgray")
ax3d = plt.gca(projection="3d") # 创建三维坐标
# ax3d.view_init(elev=70, azim=30)
ax3d.view_init(elev=45, azim=30)
#三维
plt.title('三维空间的映射', fontsize=18)
ax3d.set_xlabel('x', fontsize=20)
ax3d.set_ylabel('y', fontsize=20)
ax3d.set_zlabel('z', fontsize=20)
plt.tick_params(labelsize=10)
ax3d.set_xticks(ra)
ax3d.scatter(x, y, z, s=30, c=labels, cmap="Set1", marker="o")
plt.show()

#遗传算法特征选择部分
X=need_data
y=yy.iloc[:,0].tolist()
estimator = RFC(n_estimators=30)
model = GeneticSelectionCV(
          estimator, #模型
          cv=5, #交叉验证
          verbose=0,
          scoring="accuracy",#评价标准
```

```
            max_features=25,#最多模型特征
            n_population=100, #种群大小
            crossover_proba=0.5,#交叉概率
            mutation_proba=0.2, #变异概率
            n_generations=50,#迭代次数
            crossover_independent_proba=0.5,
            mutation_independent_proba=0.04,
            tournament_size=3,
            n_gen_no_change=10,
            caching=True,
            n_jobs=-1)
model = model.fit(X, y)
print('Features:', X.columns[model.support_])

#递归特征删除部分
score_list=[]
for index in range(7,26):
  RFR_ = RandomForestClassifier(n_estimators=30)
  selector1 = RFE(RFR_, n_features_to_select=index,
     step=1).fit(X, Y) #
     n_features_to_select表示筛选最终特征数量，step表示每次排除一个特征
  # selector1.support_.sum() # 选择特征数量
  X_wrapper1 = selector1.transform(X) #
     最优特征，返回的是筛选后的自变量
  score =cross_val_score(RFR_, X_wrapper1, Y, cv=9).mean()
  print(score)#对筛选后的指标进行随机森林回归的交叉验证的得分
  print("Support is %s" % selector1.support_) # 是否保留
  score_list.append([index,score,selector1.support_])
score_list

import xgboost as xgb
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
%matplotlib inline
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus'] = False
def schaffer(x):
  x1, x2, x3 ,x4,x5, x6, x7 ,x8,x9, x10, x11 ,x12,x13, x14,
     x15 ,x16,x17, x18, x19, \
  x20, x21, x22 ,x23,x24, x25, x26 ,x27, x28, x29, x30
     ,x31,x32, x33, x34 ,x35,x36, x37, x38 ,x39, \
  x40, x41, x42 ,x43,x44, x45, x46 ,x47, x48, x49, x50
     ,x51,x52, x53, x54 ,x55 = x
  reward = 0


  temp1 = pd.DataFrame(index=needed1.columns,data =[x19, x14,
     x55, x12, x41, x4, x11, x6, x20, x26, x18, x34, x51, x22,
     x47,
                                x33, x45,x29, x16, x17, x35,
                                   x53])
  xgbtemp1 = xgb.DMatrix(temp1.T)
```

```
    key1 = model1.predict(xgbtemp1)
    key_predictions1 = [round(value) for value in key1][0]
    if key_predictions1 == 1:
      reward = reward+1
    temp2 = pd.DataFrame(index=needed2.columns,data =[x28, x40,
       x37, x42, x38, x27, x30, x8, x6, x44, x15, x13, x22, 23])
    xgbtemp2 = xgb.DMatrix(temp2.T)
    key2 = model2.predict(xgbtemp2)
    key_predictions2 = [round(value) for value in key2][0]
    if key_predictions2 == 1:
      reward = reward+1

    temp3 = pd.DataFrame(index=needed3.columns,data =[x36, x21,
       x4, x44,x41, x52, x46])
    xgbtemp3 = xgb.DMatrix(temp3.T)
    key3 = model3.predict(xgbtemp3)
    key_predictions3 = [round(value) for value in key3][0]
    if key_predictions3 == 0:
      reward = reward+1

    temp4 = pd.DataFrame(index=needed4.columns,data =[x3, x25,
       x49, x32, x38, x39, x31])
    xgbtemp4 = xgb.DMatrix(temp4.T)
    key4 = model4.predict(xgbtemp4)
    key_predictions4 = [round(value) for value in key4][0]
    if key_predictions4 == 1:
      reward = reward+1

    temp5 = pd.DataFrame(index=needed5.columns,data =[x50, x43,
       x5, x54, x1, x10, x2, x24, x7, x9, x48])
    xgbtemp5 = xgb.DMatrix(temp5.T)
    key5 = model5.predict(xgbtemp5)
    key_predictions5 = [round(value) for value in key5][0]
    if key_predictions5 == 0:
      reward = reward+1
  return -reward
  from sko.GA import GA

  ga = GA(func=schaffer, n_dim=55, size_pop=100, max_iter=80,
     prob_mut=0.001, lb=ll, ub=uu,
      precision=pre)

  best_x, best_y = ga.run()
  print('best_x:', best_x, '\n', 'best_y:', -best_y)
  f =[]
  def func(x):
    x1, x2, x3 ,x4,x5, x6, x7 ,x8,x9, x10, x11 ,x12,x13, x14,
       x15 ,x16,x17, x18, x19, x20= x

    temp = pd.DataFrame(index=X.columns,data = [x1, x2, x3
       ,x4,x5, x6, x7 ,x8,x9, \
                                  x10, x11 ,x12,x13, x14, x15
                                     ,x16, x17,\
                                  x18, x19 ,x20])
```

```
    temp2 = xgb.DMatrix(temp.T)
    f.append(-model.predict(temp2)[0])
return -model.predict(temp2)[0]
from sko.GA import GA

ga = GA(func=func, n_dim=20, size_pop=100, max_iter=200,
    prob_mut=0.001, lb=l, ub=u,
      precision=[1e-7,1e-7,1,1e-7,1e-7,1e-7,1e-7,1e-7,1e-7,1e-7,1e-7,1e-7,1e-7,1
best_x, best_y = ga.run()
print('best_x:', best_x, '\n', 'best_y:', -best_y)
```