# 统计分析与建模作业3

小组成员：

1111111 张某某

2222222 李某某

**问题1**

1. **数据预处理：**

df=pd.read\_csv('data.csv')

df=df.drop\_duplicates()

df=df.dropna()##去除空白值和重复值

df\_zscore = df.copy() # 复制一个用来存储Z-score得分的数据框

cols = df.columns # 获得列表框的列名

for col in cols:

df\_col = df[col] # 得到每一列的值

if(col!='SMILES'):

z\_score =(df\_col - df\_col.mean()) / df\_col.std() # 计算每一列的Z-score得分

df\_zscore[col] = z\_score.abs() > 2.2 # 判断Z-score得分是否大于2.2，如果是则是True，否则为False

else:

df\_zscore[col]=False

for col in cols:

df\_drop\_outlier = df[df\_zscore[col] == False]

##处理异常值

df=df\_drop\_outlier

1. **相关性分析：**

df\_Spearman=df.corr(method='spearman')['pIC50'].abs().sort\_values(ascending=False)

df\_Spearman[1:20]

使用Spearman进行相关性分析并进行输出

|  |  |
| --- | --- |
| 特征名 | 相关系数 |
| MDEC-23 | 0.549059 |
| MLogP | 0.545199 |
| LipoaffinityIndex | 0.524914 |
| C1SP2 | 0.502404 |
| nC | 0.486774 |
| CrippenLogP | 0.473761 |
| maxsOH | 0.461918 |
| AMR | 0.454518 |
| ATSp5 | 0.451414 |
| SwHBa | 0.445723 |

**问题2**

1. **使用多元线性回归**

**1.模型概览图**

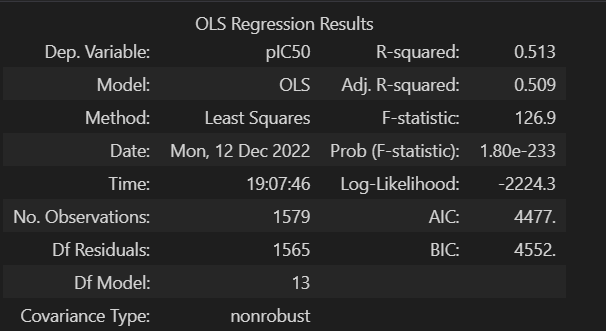
from sklearn.model\_selection import train\_test\_split

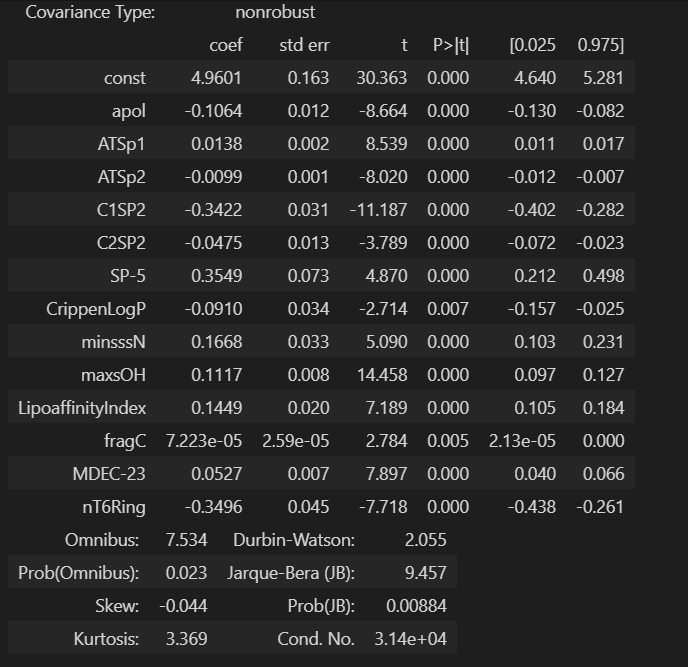
X\_train,X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=100)

y\_pIC50\_train = y\_train.loc[:,"pIC50"]

y\_pIC50\_test = y\_test.loc[:,"pIC50"]

划分测试集和训练集





**2.拟合优度**

0.509 通过上图可以看出

**3.解读回归系数**

通过上图可以看出，回归系数越大，对结果的影响也就越大。其中有些回归系数很小，可能是因为单位问题。Apol ATSp2 C1SP2 C2SP2 CrippenLogP nT6Ring对ERα的生物活性有抑制作用（相关系数为负），ATSp1 SP-5 minsssN maxsOH LipoaffinityIndex等对ERα的生物活性有促进作用。其中SP-5的促进作用最明显。

1. **使用随机森林**

**1.模型概览图**

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=100)

y\_pIC50\_train = y\_train.loc[:,"pIC50"]

y\_pIC50\_test = y\_test.loc[:,"pIC50"]

划分测试集和训练集

from sklearn.ensemble import RandomForestRegressor

rf\_model = RandomForestRegressor(n\_estimators = 250,min\_samples\_leaf = 3,max\_depth=40)

rf\_model.fit(X\_train,y\_pIC50\_train) # 预测IC50\_nM的模型训练



**2.拟合优度**

score = rf\_model.score(X\_test, y\_test)

score#拟合优度

0.7288201636660261

**问题3**

**1.随机森林模型**

1. **模型创建和ROC曲线**

cols = df.columns # 获得列表框的列名

X = df.copy()

for col in cols:

if(col not in df\_x):

X = X.drop([col],axis=1)

y = df.loc[:, ['Caco-2']]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=666)

划分训练集和测试集

from sklearn.preprocessing import StandardScaler

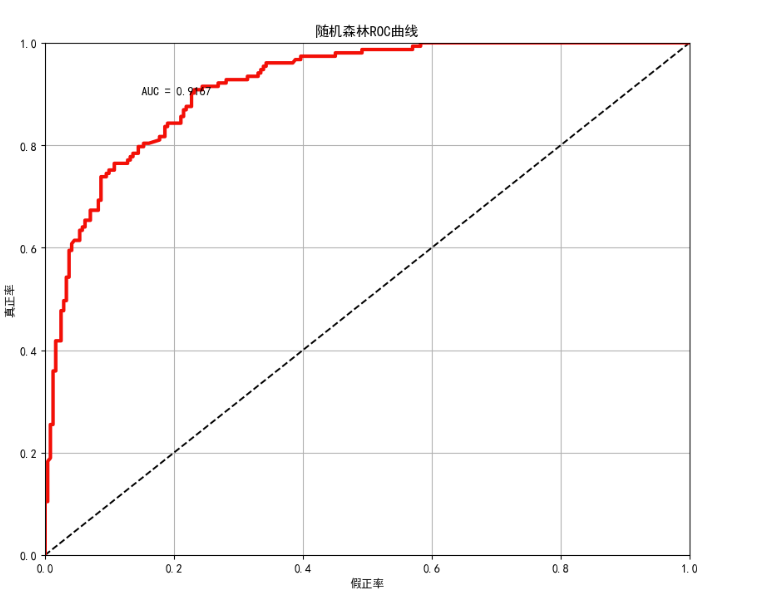
std = StandardScaler()

std.fit(X\_train)

X\_train\_std = std.transform(X\_train)

X\_test\_std = std.transform(X\_test)

创建随机森林模型



1. **模型解读**

随机森林是集群分类模型中的一种，随机森林是用随机的方式建立一个森林，森林由很多的决策树组成，且每一棵决策树之间是没有关联的。得到随机森林模型后，当新样本进入时随机森林中的每一棵决策树分别进行判断，bagging集合策略比较简单，对于分类问题通常使用投票法，得到最多票数类别或者类别之一为最终模型输出。对于回归通常使用简单平均法，T个弱学习器得到的回归结果进行算术平均即最终模型输出

1. **质量评估**

from sklearn.metrics import confusion\_matrix, f1\_score, accuracy\_score

as\_s=accuracy\_score(y\_test, y\_predicted)

f1\_s=f1\_score(y\_test,y\_predicted)

print(as\_s)

print(f1\_s)

**准确率** 0.810126582278481

**F1分数** 0.7747747747747747

AUC 0.9167

**结果尚且令人满意**

**2.逻辑回归模型**

1. **模型创建和ROC曲线**

from sklearn.model\_selection import train\_test\_split

cols = df.columns # 获得列表框的列名

X = df.copy()

for col in cols:

if(col not in df\_x):

X = X.drop([col],axis=1)

y = df.loc[:, ['Caco-2']]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=666)

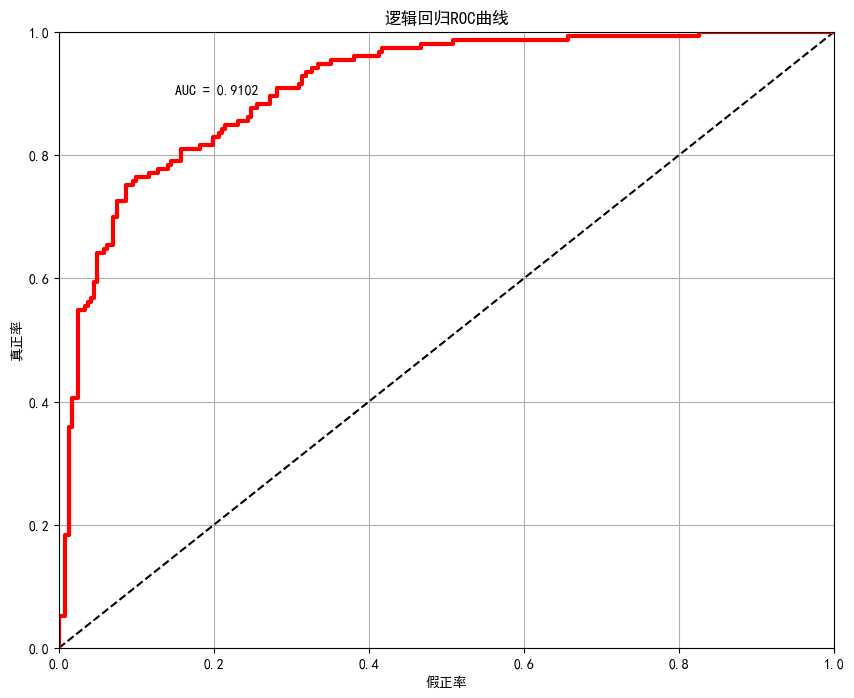
划分训练集和测试集

from sklearn.linear\_model import LogisticRegression

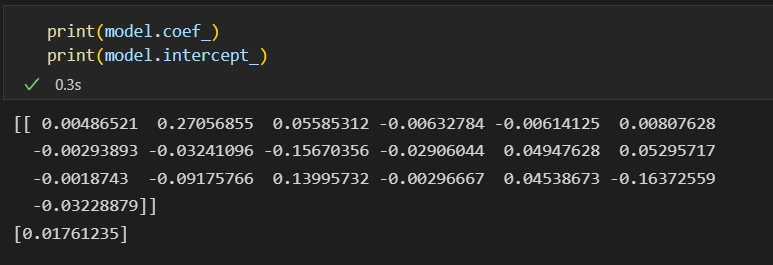
model = LogisticRegression()

model.fit(X\_train,y\_train)

创建逻辑回归模型



1. **模型解读**



1. **质量评估**

from sklearn.metrics import confusion\_matrix, f1\_score, accuracy\_score

as\_s=accuracy\_score(y\_test, y\_predicted)

f1\_s=f1\_score(y\_test,y\_predicted)

print(as\_s)

print(f1\_s)

**准确率** 0.8354430379746836

**F1分数** 0.7854785478547854

AUC 0.9102

**结果尚且令人满意**