基于决策树算法的 CO_2 吞吐模型研究

摘 要

在实际石油勘测与采油过程中，利用注入 CO_2 的方式进行开采的方式是工程实际上常用的一种方法。但是 CO_2 吞吐效果受到多种因素影响，需要构建相应模型分析不同因素变化对效果的影响。基于此，本文使用决策树算法对问题进行建模与分析。

对于问题一的特征筛选，使用特征工程中的常见手段，对比传统统计学方法并通过构建决策树作为预训练模型为每个属性赋予相应重要性分数，从而选择出合适的五项指标用于学习。

对于问题二的阈值确定，以 450 为阈值将 CO_2 吞吐增油量划分为二值变量，该问题也被转化为一个二分类问题。而对于这一二分类问题，通过构建决策树分类器同样可以对指标阈值进行划分与确定，准确度可以达到 0.9029。另外，从灰色关联分析和相应领域研究经验两个角度可以发现，这五项指标与目标是高度关联的，而且也得到了工程实际上的论证。

对于问题三的预测问题，构建了决策树回归器，模型效果表现良好。与实际值相比，预测值与实际的 R2 分数可以达到 0.96，属于高度相关的数值。

模型对 CO_2 吞吐效果进行相应构建与研究，效果上表现准确而且相对稳定，在实际工程应用中有一定参考价值。

关键词：CO_2 吞吐模型，决策树算法，灰色关联分析，特征工程

§一　问题的重述

一、背景知识

CO_2 吞吐可以提高单井产油量，CO_2 吞吐效果一般用指标“增油量”表征。现有数据证明，增油量受多种因素影响，且不同井的 CO_2 吞吐增油量差异很大，因此需要建立一套适用于 CO_2 吞吐的选井标准，并进行 CO_2 吞吐效果预测。

二、问题描述

附件 1 提供了 17 种因素对应的 CO_2 吞吐增油量。其中因素 A~L 为选井因素，是不能通过人为操作更改的因素；因素 M~Q 为工艺因素，即可通过人为操作更改的因素。

（1）根据附件 1 中所给的数据，在选井因素中选取不多于五类的 CO_2 吞吐效果敏感选井因素。

（2）根据附件 2 中人工边水驱筛选标准的形式，以增油量 450m_3 为下限，建立基于敏感选井因素的 CO_2 吞吐选井标准，并论证选井标准的合理性。

（3）利用问题二中的敏感选井因素和附件一的工艺因素，建立 CO_2 吞吐增油量的预测模型，并论证预测模型的精度。

§二　问题的分析

1 问题一的分析

问题一是一个典型的特征工程问题。在附件 1 的 17 组数据中，我们运用机器学习方法，利用决策树模型计算信息增益来给每个特征一个对应的分数，分数越高特征影响越强。根据训练出的决策树模型筛选的指标，做出变量因素和增油量相关性对应分数，根据分数值的大小筛选出对 CO2 吞吐效果敏感选井因素。

2 问题二的分析

对于问题二的阈值设置，我们同样通过构建决策树，以 450 为阈值将目标划分为二值变量从而构建基于决策树的分类器，并通过可视化方法展示出每个属性对应的阈值。为验证属性选择的合理性，我们从灰色关联性分析和相关领域知识两个角度出发对结果进行了探讨。

3 问题三的分析

问题三要求我们建立 CO2 吞吐增油量的预测模型，并论证预测模型的精度，这里需要我们基于对问题一和问题二的算法的分析之后，建立回归预测模型求解程序运行效率最高的方案。这里我们同样基于决策树进行，训练一个回归树进行求解。

§三　符号说明

三、符号说明

序	符号	意义
1	x	一列数据
2	μ	数据均值
3	σ	数据的标准差
4	r	相关系数
5	d	数据的秩次
6	n	数据的个数
7	Gain	信息增益
8	GainRatio	信息增益率
9	Ent	信息熵
10	p	属性的某一取值在数据集中的占比
11	Gini	基尼值
12	GiniIndex	基尼指数
13	ξ	灰色关联指数

§四　模型的建立与求解

1. 问题一的模型建立与求解

1.1 特征工程

问题一要求我们筛选出影响最强的五个指标，是一个典型的特征工程问题。我们将原始数据的所有属性数值化以后得到了数据的原始特征，而原始特征需要进行处理与筛选以后才能为我们所进一步使用。正确的特征应该适合当前的任务，并易于被模型所使用。特征工程就是在给定数据、模型和任务的情况下设计出最合适的特征的过程[1]。

特征工程的第一步是查看数据的属性特征，我们将原始数据的部分属性统计特征列在表 1 中。可以看到，数据条数不多，而且很多特征分布非常集中，差异不大。

表 1. 原始数据的部分属性特征

统计量	地层倾角	油层有效厚度/m	非均质（洛伦兹系数）	投产含水/f	产水段非均质程度（渗透率倍数）	平行井采液速度/rm3/d	吞吐时机/f	周期注气量/sm3	注气速度/sm3/d	关键时间/d	开井后采液速度/rm3/d	CO2吞吐增油量

count	175	175	175	175	175	175	175	175	175	175	175	175
mean	3.057142857	6.606285714	0.497142857	0.1056	17.42285714	6	0.931142857	151714.2857	50000	29.95428571	10.2	452.7538778
std	0.574499139	0.741166701	0.049629167	0.136097328	18.99587199	37.61572183	0.107017332	20690.06568	4673.230195	3.082798294	2.418796129	116.0073824
min	3	1.1	0	0	1	0	0.2	100000	20000	7	5	3.09967
25%	3	6.6	0.5	0	5	0	0.95	150000	50000	30	10	459.62
50%	3	6.6	0.5	0	5	0	0.95	150000	50000	30	10	478.3334
75%	3	6.6	0.5	0.28	15	0	0.95	150000	50000	30	10	489.01
max	10	13.2	0.8	0.28	50	350	0.95	400000	100000	60	40	1088.3334

在查看原始数据的重要统计特征以后，我们基于此进行第二步操作，即原始数据的预处理操作。这里可以把平行井距离中“无平行井”视作缺失数据，采取常数填充法，填充-1 作为空缺值的代替值。这样我们就完成了缺失值的填充工作。

对于离散数据，我们采用标签法，即对不同的类别给予不同的整数标记，实现从文本到数值的转化。而对于连续数值，我们使用标准化和归一化两步操作。其中，标准化的表达式为：

$$x = \frac{x - \mu}{\sigma} \quad (1)$$

而对于归一化，表达形式为：

$$x = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2)$$

经过预处理以后，数据已经被映射归约到一个有限的范围，这时我们进行特征选择的操作。特征选择通常由三种方法构成：第一种方法是针对每个变量进行方差的计算，设置阈值，低于一定阈值说明该属性变化不大，分布较为集中；第二种方法是针对属性与目标间的相关性或独立性检验。由于目标为连续变量，属性中的离散变量存在多个取值，故并不适合进行双样本 t 检验。考虑到数据已经数值化，可以使用相关系数检验法分析属性与目标的关联性。

利用方差检验得到排名前五的指标分别为隔夹层位置，投产含水/f，产水段长度，

产水段位置和产水段非均质程度（渗透率倍数）。但主要问题在于，原始数据集的属性分布均较为集中，可能存在某一属性虽然取值较为集中但一旦改变则容易引起目标的剧变。故方法一并不严格适用。

对于方法二中提到的相关系数检验法，若两列数据均呈正态分布且大致线性则可以使用皮尔逊相关系数：

$$r=\frac{\sum_{i=1}^{n}x_iy_i-n\bar{x}\bar{y}}{\sqrt{\sum_{i=1}^{n}x_i^2-n\bar{x}^2}\sqrt{\sum_{i=1}^{n}y_i^2-n\bar{y}^2}} \quad (3)$$

但若数据不满足先决条件，则改用斯皮尔曼相关系数：

$$r=1-\frac{6\sum_{i=1}^{n}d_i^2}{n(n^2-1)} \quad (4)$$

通过相关系数法得到的前五名指标为隔夹层位置，投产含水/f，产水段非均质程度（渗透率倍数），吞吐时机/f，周期注气量/sm3。但这一方法存在一个问题，在于其中很多数据是离散变量数值化以后求解的相关系数，可能会有一定偏差。对于多取值的离散变量，使用二值化的 t 检验和纯粹连续的相关系数法均并非最优方案。

这时我们将目光投到方法三上，使用预训练的机器学习模型对特征进行选择。

1.2 决策树算法

由于数据中离散属性居多，我们考虑使用基于决策树的模型进行求解。

决策树是一类常见的机器学习方法，基于树结构进行决策。决策树由一个根节点，若干个中间节点和若干个叶子节点构成，其节点的划分基于信息熵与信息增益，是递归划分的模式[2]，基本学习算法如图：

Input: 训练集 $D=(x_1,y_1),(x_2,y_2),...,(x_n,y_n)$
属性集 $A=a_1,a_2,...,a_d$
过程:决策树生成
生成结点node;
if D中样本全部属于一个类别 **then**
  将node标记为C类叶子节点;
  return;
end
if A是个空集 *or* D样本在A中取值相同 **then**
  将node标记为叶子节点，类别标记为D中样本数最多的类;
  return;
end
从属性集里面选择最优划分属性a_0，这是各种决策树优化的关键
for a_0^v *in* a_0
do
  为node生成一个分支;
  令D_v表示D中在a_0上取值为a_0^v的样本子集;
  if D_v是个空集 **then**
    将分支节点标记为叶子节点，类别标记为D中样本最多的类;
    return;
  end
  else
    以TreeGenerate(D_v,$A-\{a_0\}$)为节点递归生成
  end
end
Output: 以node为根节点的一棵决策树

图 1. 决策树节点划分生成的伪代码

在这个递归过程中有三种典型情况导致递归返回：(1). 当前节点包含的样本属于同一样本；(2). 属性集为空集，或者所有样本在所有属性上取值相同无法划分；(3). 当前节点包含样本集合为空[3]。而从图 1 中可以看出，算法的关键在于如何选择最优划分属性的标准。事实上，这也正是三类典型决策树的区别之处：

对于 ID3 决策树，使用对某属性划分得到的信息增益描述节点[4]：

$$Gain(D,a)=Ent(D)-\sum_{v=1}^{V}\frac{|D^v|}{|D|}Ent(D^v) \quad (5)$$

而改进以后的 C4.5 决策树使用增益率作为描述节点的标准[5]：

$$GainRatio(D,a)=\frac{Gain(D,a)}{-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\log_2\frac{|D^v|}{|D|}} \quad (6)$$

这两类决策树的最大局限是只能处理分类问题而无法处理回归问题。对于此问题，目标为连续数值的话需要使用分类回归树(Classification and Regression Tree, CART)进行处理[6]。CART 以基尼指数作为划分标准：

$$\begin{cases} Gini(G)=1-\sum_{k=1}^{|y|}p_k^2 \\ GiniIndex(D,a)=\sum_{v=1}^{V}\frac{|D^v|}{|D|}Gini(D^v) \end{cases} \quad (7)$$

接下来，我们通过构建 CART 作为预训练模型对特征进行筛选。

1.3 基于决策树的特征选择

基于树结构的机器学习模型通常会通过信息增益或基尼指数划分标准作为属性的重要性从而为每个属性赋予相应的权重，这也是我们最终采用的特征选择方法。

构建一个 CART 学习器，在数据集上以训练集：测试集=7：3 的比例反复交叉验证 100 次以后取每个属性权重的平均值作为最终权重，我们将最终结果绘制在图 2 中：

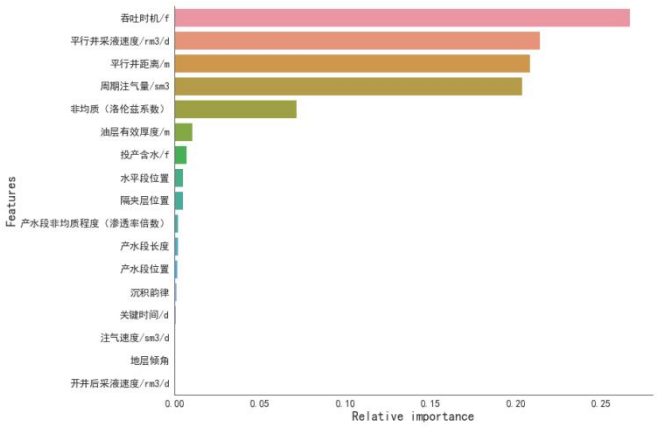

图 2. 基于决策树的指标筛选结果

最终我们得到的排名前五的指标为：油层有效厚度/m，非均质（洛伦兹系数），平行井采液速度/rm3/d，吞吐时机/f，周期注气量/sm3。

2. 问题二的模型建立与求解

2.1. 决策树用于分类问题

同样基于决策树方法，若将目标按照 450 作为阈值，高于 450 者记为 0 而低于 450 者记为 1，则完成了目标的二值化，并将问题转化为一个二分类问题。对于此类二分类问题，我们同样基于决策树构建分类器，并获取前五名的属性的阈值大小。

一个有趣的现象是，当我们使用全部数据进行训练时，我们得到了两棵结构不同的决策树，而这两棵决策树在数据集上的分类效果竟然是一样的。如图 3 所示：

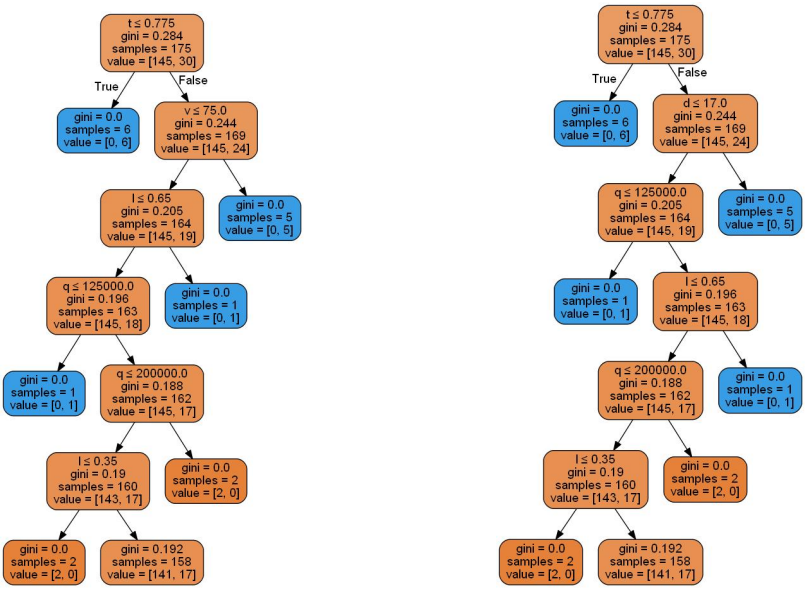

图 3. 训练得到的两棵分类树，蓝色节点表示明确为低于 450 者

这两棵树的分类结果均如表 2 所示，我们基于其预测结果绘制了 AUC 曲线如图 4 所示，虽然准确率达到了 0.9029 但 AUC=0.7245 说明我们的模型还存在一定不稳定性。

表 2. 分类器的分类结果

statistic	precision	recall	f1-score	support
>450	0.9	1	0.94	145
<450	1	0.43	0.6	30
accuracy			0.9029	175
macro avg	0.95	0.72	0.77	175
weighted avg	0.91	0.9	0.89	175

最终的阈值结果如表 3 所示。接下来我们将对指标和阈值的合理性展开进一步的讨论与建模。

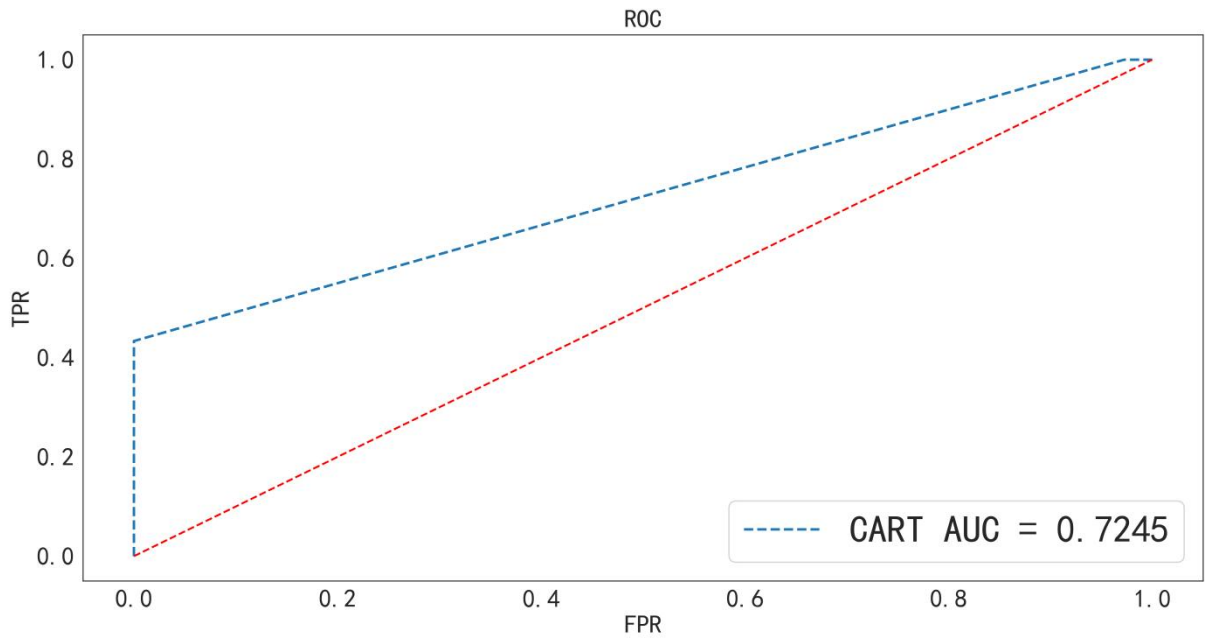

图 4. 分类器分类过程中的 AUC 曲线

表 3. 各指标的阈值结果

筛选条件	筛选要求
吞吐时机	大于 0.775
平行井采液速度	平行井采液速度小于等于 75.0
平行井距离	平行井距离小于等于 17.0
周期注气量	周期注气量大于 200000、周期注气量不大于 200000 且非均质（洛伦兹系数）不大于 0.35
非均质（洛伦兹系数）	非均质（洛伦兹系数）不大于 0.65

2.2. 灰色关联分析对指标合理性的讨论

为检验选取指标的合理性，考虑到数据集中数据条目不多，我们采用灰色关联分析的方式对指标之间的关联性进行建模。

灰色关联分析方法，是根据因素之间发展趋势的相似或相异程度，亦即“灰色关联度”，作为衡量因素间关联程度的一种方法[7]。其思想很简单，确定参考列和比较列以后需要对数列进行无量纲化处理，然后计算灰色关联系数。这里我们使用均值处理法，即每个属性的数据除以对应均值：

$$x(i)=\frac{x(i)}{\overline{x}(i)} \quad (8)$$

灰色关联系数的定义如下：

$$\xi_i(k)=\frac{\min\limits_s\min\limits_t|x_0(t)-x_s(t)|+\rho\max\limits_s\max\limits_t|x_0(t)-x_s(t)|}{|x_0(t)-x_i(t)|+\rho\max\limits_s\max\limits_t|x_0(t)-x_s(t)|} \quad (9)$$

其中 ρ 不超过 0.5643 时分辨力最好，这里为了简洁，取之为 0.5。灰色关联度为关联系数在样本上的平均值，计算出每个属性的灰色关联度以后我们进行分析，得到的五个属性灰色关联度分别如表 4 所示：

表 4. 五个属性的灰色关联度

属性	灰色关联度
吞吐时机/f	0.998936
非均质（洛伦兹系数）	0.998489
周期注气量/sm3	0.998144
平行井采液速度/rm3/d	0.980787
平行井距离/m	0.951890

可以看到，这五项指标和目标的灰色关联系数都比较高，说明其影响确实比较大。

2.3. 结果在能源化工领域的合理解释

对于吞吐时机，相同 CO2 注入量时，注入时机越早，增油量越多，换油率越高。注气时机应选择油井投产初期，此时近井地层原油相态特征有利于 CO2 吞吐增产机理实现[8]。而此处吞吐时机单位并不是常用的时间单位，当数值越大表明注入越早。

对于非均质（洛伦兹系数），经验表明储层的非均质性对 CO2 吞吐有着显著影响，一般非均质系数小于 0.6 的油藏实施 CO2 吞吐的成功率较高[9]。

对于周期注气量，周期注气也可以大幅提高采收率，且在一定程度上也延缓了气窜的发生，主要原因是:周期性的注采在地层中造成了不稳定的压力场,使得地层油水在岩心孔隙中不断重新分布[10]；岩心中不同渗透率区域之间压力传导系数不同，且高渗区导压系数较高,低渗区较低,低渗区域压力传导较慢,形成的反向压力差促使低渗区域的部分原油被排出，从而提高了采收率[11]

对于平行井采液速度，采液速度越高,油井生产过程中含水越高,含水上升越快,含水上升率越大。过快的含水上升速度会使油井产水量成倍增加,增加采油成本,同时,油井会被快速水淹,降低了可采储量及最终采收率，从统计结果来看,采液速度与采收率呈反相关关系,采液速度越低采收率越高[12]。

对于平行井距离，水平井井组 CO2 吞吐具有明显的协同增油效果.相同实验条件下,高部位井注气吞吐起到气顶作用,中,低部位井含水率降低不明显,边水的抑制效果不明显,井组增油量低,气体利用率最低;中部位井注气吞吐控水增油效果介于二者之间，所以存在一个平衡的临界值[13]。

3. 问题三模型的建立与求解

3.1 分类回归树

对于问题三的预测模型，我们延续决策树的思想，构建 CART 模型完成回归任务。

CART 分类回归树是一种典型的二叉决策树，可以做分类或者回归。如果待预测结果是离散型数据，则 CART 生成分类决策树；如果待预测结果是连续型数据，则 CART

生成回归决策树。数据对象的属性特征为离散型或连续型，并不是区别分类树与回归树的标准。作为分类决策树时，待预测样本落至某一叶子节点，则输出该叶子节点中所有样本所属类别最多的那一类（即叶子节点中的样本可能不是属于同一个类别，则多数为主）；作为回归决策树时，待预测样本落至某一叶子节点，则输出该叶子节点中所有样本的均值。对于这一回归问题，其核心的目标函数为：

$$J = \min_{j,s}\left[\min_{c_1}\sum_{x_i \in R_1(j,s)}(y_i - c_1)^2 + \min_{c_2}\sum_{x_i \in R_2(j,s)}(y_i - c_2)^2\right] \quad (10)$$

将数据集按照 7：3 的比例随即切分为训练集与测试集，在训练集上训练模型并在测试集上进行测试，限制决策树最大深度为 5，得到结果的 R2 分数为 0.9629；若不对深度加以限制，则 R2 分数为 0.9643，说明深度的增加并不会对结果产生显著影响。我们将这一回归树的结构可视化为图 5：

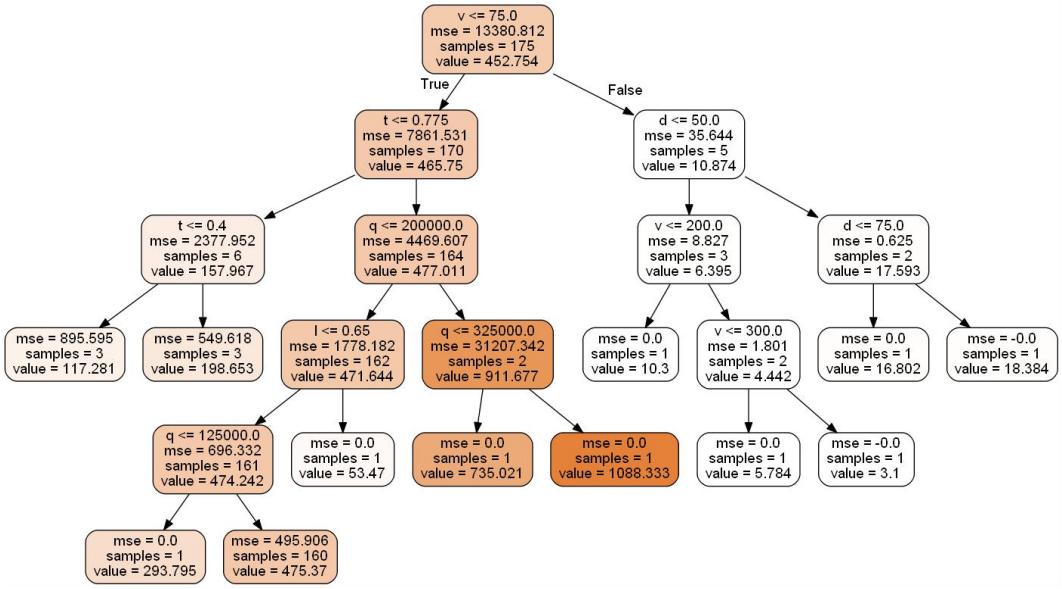

图 5. 构建回归树的结构

相比于神经网络的 R2=0.43 和线性回归的 0.62，这一结果很显然表现更优。

3.2 对结果的讨论

对数据进行预测以后，我们将原始值和预测值的箱线图绘制如图 6 所示：

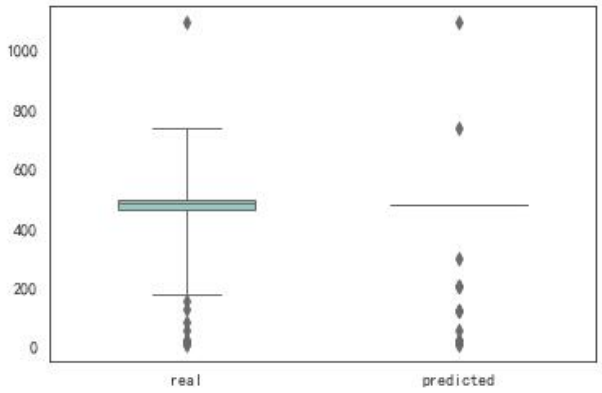

图 6. 真实值和预测值的箱线图

很明显可以看到，相比于真实值，预测数值分布更加集中但偶有离群点。二者均值接近，最大值最小值也接近，并且在偏大值偏小值占比不大的情况下这类模型可以表现出良好的特征，取得良好效果。

我们将实际值与预测值的偏差绘制如图 7 所示，其中横坐标为样本，纵坐标表示偏差，可以看到，数值偏差并不明显，偶有少数样本分类偏差较大，大部分偏差都在 0 附近，所以这一模型表现实际上已经到了一个比较良好的水准。

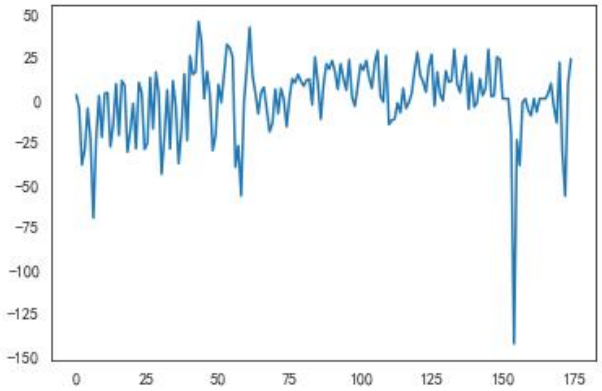

图 7. 真实值与预测值的偏差图

§ 五 模型的优缺点与改进

1．优点：

1. 基于决策树算法完成特征工程，机器学习计算得到的模型相比于带有一定主观因素（方差或相关系数的阈值选择）的纯粹统计方法更有说服力。

2. 基于决策树算法的属性阈值选择通过将原问题转化为一个二分类问题构建分类决策树，使得各变量的取值被结构化地组织在一起，能够清晰地展示出来。

3. 决策树算法在本问题中取得了很良好的表现，无论是分类问题还是回归问题都有很好的准确性。

4. 这一算法的思想能够被迁移到其它问题上，而且模型结构能够被清晰地可视化。

2．缺点：

1. 在分类问题上，这一算法仍然没有表现出足够的鲁棒性，可以说很大程度上受到了数据自身的影响。

2. 阈值组合的说理还不够明确，还需要在相应专业领域得到进一步实验论证。

3. 改进方案设计：

1. 尝试使用鲁棒性更好的 GBDT 框架例如 XGBoost, LightGBM 等算法

2. 在分类器设计问题上尝试使用过采样的方法调整类别不平衡的比例

3. 对比神经网络与支持向量机等其它非树结构算法观察效果

参考文献

[1] Alice Zheng, Amanda Casari, Feature Engineering For Machine Learning,2018

[2] 李航. 统计学习方法[M]. 清华大学出版社, 2012.

[3] 周志华. 《机器学习》[J]. 中国民商, 2016, 03(No.21):93-93.

[4] Peng H . A Study on Internet-Based Autonomous Learning of College Students' English Based on ID3 Algorithm[M]. 2021.

[5] Ahmad M , Al-Shayea N , Tang X , et al. Predicting the Pillar Stability of Underground Mines with Random Trees and C4.5 Decision Trees[J]. Applied Sciences, 2020, 10(6486):1-12.

[6] Breiman L , etc. Classification and regression trees[M].

[7] 邓聚龙. 灰色系统[M]. 国防工业出版社, 1985.

[8] 刘刚.致密油体积压裂水平井 CO_2 吞吐注采参数优化[J].石油地质与工程,2020,34(02):90-93.

[9] 左翼. 致密油藏 CO_2 吞吐适应性评价方法及参数优化研究[D].中国石油大学(北京),2018.

[10] 王维波,陈龙龙,汤瑞佳,王贺谊,杨红.低渗透油藏周期注 CO_2 驱油室内实验[J].断块油气田,2016,23(02):206-209.

[11] 王维波,师庆三,余华贵,黄春霞,陈龙龙.二氧化碳驱油注入方式优选实验[J].断块油气田,2015,22(04):497-500+504.

[12] 屈啸,邱坤态,马培申.强边水断块油藏合理采液速度研究[J].石化技术,2020,27(09):154+156.

[13] 王志兴, 赵凤兰, 侯吉瑞,等. 断块油藏水平井组 CO2 协同吞吐效果评价及注气部位优化实验研究[J]. 石油科学通报, 2018(2).

附录

环境：Intel i7+NVIDIA GEFORCE GTX 1650+Anaconda 下的 Python3.7.8+VS Code

代码 1. 特征工程

```
# -*- coding: utf-8 -*-
"""
Created on Sat Jun 12 08:59:54 2021

@author: mengmeng
"""
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
#%matplotlib inline
sns.set_style("white")
plt.rcParams['font.sans-serif']=['Simhei']
plt.rcParams['axes.unicode_minus']=False
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder
from scipy.stats import kstest
data=pd.read_excel("CO2.xlsx")
data=data.drop(['Unnamed: 0'],axis=1)
le = LabelEncoder()
standard_scaler  = preprocessing.MinMaxScaler()
#特征工程第一步：数据预处理
#针对离散变量的标签
catagories=['隔夹层位置','沉积韵律','产水段长度','产水段位置','水平段位置']
for cat in catagories:
    t=le.fit_transform(data[cat].astype(str))
    data[cat]=t
#这一个属性有些特殊
dis=[]
for items in data['平行井距离/m']:
    if items=='无平行井':
        dis.append(-1)
    else:
        dis.append(items)
data['平行井距离/m']=dis
#数值变量的归一化
numbers=['地层倾角','油层有效厚度/m','非均质（洛伦兹系数）','投产含水/f','产水段非均质程度（渗透率倍数）',\
```

```
        '平行井采液速度/rm3/d','吞吐时机/f','周期注气量/sm3','注气速度/sm3/d','关键时间/d',\
            '开井后采液速度/rm3/d','CO2 吞吐增油量']
max_tar=data['CO2 吞吐增油量'].max()
min_tar=data['CO2 吞吐增油量'].min()
for num in numbers:
    data[num]=(data[num]-data[num].min())/(data[num].max()-data[num].min())

#特征工程第二部：特征指标筛选
#统计学方法
#实验一：分析每个变量的方差
Std=[]
for num in data.columns:
    t=(data[num]-data[num].min())/(data[num].max()-data[num].min())
    Std.append(t.std())
feature1=sorted(enumerate(Std), key=lambda x:x[1])
#统计得到的特征为隔夹层位置，投产含水/f，产水段长度 ，产水段位置，产水段非均质程度（渗透率倍数）
'''
实验结果分析：
这一结果是单纯看数据分布的偏差。但是方法似乎并不合理，因为有的指标可能没什么偏差但一旦改变可能对目标
产生很大的影响，所以要联系起来看
'''
#实验二：相关性分析
#对于离散变量使用 t 检验似乎不太好弄，因为不是二变量。不过我们已经连续化了，所以可以试试皮尔逊相关系数
Corr=[]
for fea in data.columns[:-1]:
    if kstest(data[fea],'norm').pvalue<0.05:
        t=data[fea].corr(data['CO2 吞吐增油量'],method='pearson')
    else:
        t=data[fea].corr(data['CO2 吞吐增油量'],method='spearman')
    Corr.append(t)
feature2=sorted(enumerate(Corr), key=lambda x:x[1])
#统计得到的特征为隔夹层位置，投产含水/f，产水段非均质程度（渗透率倍数），吞吐时机/f，周期注气量/sm3
'''
实验结果分析：
双样本的检验不能盲目使用皮尔逊相关系数。如果数据不服从正态分布，需要改用斯皮尔曼相关系数。
即便如此，相关系数高也不见得一定是重要的，需要确定一个阈值。
'''
#有些变量虽然方差不大但是一旦改变影响很大，单纯看数据的分布情况恐怕还不行
#实验三：使用机器学习算法暗箱生成分数
#这个时候我们尝试使用机器学习方法进行解释，最为合理
#事实上，每个监督学习的算法都可以对特征给一个分数作为权重
from sklearn import tree
cart=tree.DecisionTreeRegressor()
X=data[data.columns[:-1]]
Y=data['CO2 吞吐增油量']
```

```
cart.fit(X,Y)
importance =cart.feature_importances_
feature3=sorted(enumerate(importance), key=lambda x:x[1])
#筛选的特征为油层有效厚度/m，非均质（洛伦兹系数），平行井采液速度/rm3/d，吞吐时机/f，周期注气量/sm3
'''
结果分析：一次性将所有数据利用完毕，未进行重复实验
模型：决策树
'''
#实验四：重复实验下的机器学习方法筛选
#担心一次测试会使得结果有误差，进行多次采样交叉验证
def features_selection(X_data,Y_data,x_col):
    import_feature = pd.DataFrame()
    import_feature['col'] = x_col
    import_feature['xgb'] = 0
    # Repeat test 100 times
    for i in range(100): # 50,150
        x_train, x_test, y_train, y_test = train_test_split(X_data, Y_data, test_size=0.2, random_state
=i)

        # Define model hyper-parameters
        model = tree.DecisionTreeRegressor()

        # Model fitting
        model.fit(x_train, y_train)
        import_feature['xgb'] = import_feature['xgb']+model.feature_importances_/100

    # Sort descending
    import_feature = import_feature.sort_values(axis=0, ascending=False, by='xgb')
    print('All features:')
    print(import_feature.head(17))
    # Sort feature importances from GBC model trained earlier and locate it
    indices = np.argsort(import_feature['xgb'].values)[::-1]

    Num_f = 17
    indices = indices[:Num_f]
    # Visualise these with a barplot

    #plt.subplots(figsize=(12, 10))
    fig=plt.figure(figsize=(12,10))
    # g = sns.barplot(y=list(name_dict.values())[:Num_f], x = import_feature.iloc[:Num_f]['xgb'].values
[indices], orient='h') #import_feature.iloc[:Num_f]['col'].values[indices]
    g = sns.barplot(y=import_feature.iloc[:Num_f]['col'].values[indices], x = import_feature.iloc[:Num_
f]['xgb'].values[indices], orient='h') #import_feature.iloc[:Num_f]['col'].values[indices]
    g.set_xlabel("Relative importance",fontsize=18)
    g.set_ylabel("Features",fontsize=18)
```

```
    g.tick_params(labelsize=14)
    sns.despine()


    plt.show()
features_selection(X,Y,X.columns)
```

代码 2. 阈值确定

```
# -*- coding: utf-8 -*-
"""
Created on Sat Jun 12 18:33:12 2021

@author: mengmeng
"""
import numpy as np
import pandas as pd
from sklearn.metrics import confusion_matrix,accuracy_score,roc_curve,auc,precision_recall_curve,classi
fication_report
data=pd.read_excel("CO2.xlsx")
data=data.drop(['Unnamed: 0'],axis=1)
data=data[['吞吐时机/f','平行井采液速度/rm3/d','平行井距离/m','周期注气量/sm3','非均质（洛伦兹系数）','CO2 吞
吐增油量']]
data.replace("无平行井", -1, inplace=True)
from matplotlib import pyplot as plt
import seaborn as sns
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
#%matplotlib inline
sns.set_style("white")
plt.rcParams['font.sans-serif']=['Simhei']
plt.rcParams['axes.unicode_minus']=False
tar=[]
for i in data['CO2 吞吐增油量']:
    if i>450:
        tar.append(0)
    else:
        tar.append(1)
data['tar']=tar
from sklearn import tree
from sklearn.metrics import confusion_matrix,accuracy_score
import xgboost as xgb
import pydotplus
from IPython.display import Image,display
X=data[['吞吐时机/f','平行井采液速度/rm3/d','平行井距离/m','周期注气量/sm3','非均质（洛伦兹系数）']]
y=data.tar
from sklearn.model_selection import train_test_split
```

```
#train regressor
'''
dtree= xgb.XGBClassifier(
        max_depth=7,
        n_estimators=1,
    )
'''
dtree=tree.DecisionTreeClassifier()
dtree.fit(X,y)
y_predict=dtree.predict(X)
y_predict_proba=dtree.predict_proba(X)
#evaluate
print(classification_report(y,y_predict))
#xgb.plot_tree(dtree)
#plt.show()
dot_data=tree.export_graphviz(dtree,
                out_file=None,
                feature_names=['t','v','d','q','l'],
                #class_names=y,
                filled=True,
                rounded=True,
                special_characters=True
                )
graph=pydotplus.graph_from_dot_data(dot_data)
display(Image(graph.create_png()))
def plot_roc(labels, predict_prob,Moodel_name_i,fig,labels_name,k):
    false_positive_rate,true_positive_rate,thresholds=roc_curve(labels, predict_prob)
    roc_auc=auc(false_positive_rate, true_positive_rate)
    #plt.figure()
    line_list = ['--','-']
    ax = fig.add_subplot(111)
    plt.title('ROC', fontsize=20)
    ax.plot(false_positive_rate, true_positive_rate,line_list[k%2],linewidth=1+(1-k/5),label=Moodel_nam
e_i+' AUC = %0.4f'% roc_auc)
    plt.xticks(fontsize=20)
    plt.yticks(fontsize=20)
    plt.ylabel('TPR', fontsize=20)
    plt.xlabel('FPR', fontsize=20)
    labels_name.append(Moodel_name_i+' AUC = %0.4f'% roc_auc)
    #plt.show()
    return labels_name
fig = plt.figure(dpi=400,figsize=(16, 8))
labels_names=[]
plot_roc(y, y_predict_proba[:,1],"CART",fig,labels_names,0)
plt.plot([0,1],[0,1],'r--')
```

```
plt.legend(loc='lower right', fontsize=30)
```

代码 3. 灰色关联

```
# -*- coding: utf-8 -*-
"""
Created on Sat Jun 12 19:54:46 2021

@author: mengmeng
"""
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
#%matplotlib inline
sns.set_style("white")
plt.rcParams['font.sans-serif']=['Simhei']
plt.rcParams['axes.unicode_minus']=False
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder
from scipy.stats import kstest
data=pd.read_excel("CO2.xlsx")
data=data.drop(['Unnamed: 0'],axis=1)
le = LabelEncoder()
standard_scaler  = preprocessing.MinMaxScaler()
#特征工程第一步：数据预处理
#针对离散变量的标签
catagories=['隔夹层位置','沉积韵律','产水段长度','产水段位置','水平段位置']
for cat in catagories:
    t=le.fit_transform(data[cat].astype(str))
    data[cat]=t
#这一个属性有些特殊
dis=[]
for items in data['平行井距离/m']:
    if items=='无平行井':
        dis.append(-1)
    else:
        dis.append(items)
data['平行井距离/m']=dis
#数值变量的归一化
numbers=['地层倾角','油层有效厚度/m','非均质（洛伦兹系数）','投产含水/f','产水段非均质程度（渗透率倍数）',\
         '平行井采液速度/rm3/d','吞吐时机/f','周期注气量/sm3','注气速度/sm3/d','关键时间/d',\
             '开井后采液速度/rm3/d','CO2 吞吐增油量']
```

```
max_tar=data['CO2 吞吐增油量'].max()
min_tar=data['CO2 吞吐增油量'].min()
for num in numbers:
    data[num]=(data[num]-data[num].min())/(data[num].max()-data[num].min())

for col in data.columns:
    data[col]/=data[col].mean()

cp=data[['吞吐时机/f','平行井采液速度/rm3/d','平行井距离/m','周期注气量/sm3','非均质（洛伦兹系数）']]
ck=data['CO2 吞吐增油量']
for cols in cp.columns:
    cp[cols]-=ck

cp=cp.T
#求最大差和最小差
mmax=cp.abs().max().max()
mmin=cp.abs().min().min()
rho=0.5
#3、求关联系数
ksi=((mmin+rho*mmax)/(abs(cp)+rho*mmax))
#4、求关联度
r=ksi.sum(axis=1)/ksi.columns.size
#5、关联度排序，得到结果 r3>r2>r1
result=r.sort_values(ascending=False)
```

代码 4. 预测回归

```
# -*- coding: utf-8 -*-
"""
Created on Sat Jun 12 20:18:48 2021

@author: mengmeng
"""
import numpy as np
import pandas as pd
import pydotplus
from IPython.display import Image,display
from matplotlib import pyplot as plt
import seaborn as sns
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
#%matplotlib inline
sns.set_style("white")
plt.rcParams['font.sans-serif']=['Simhei']
plt.rcParams['axes.unicode_minus']=False
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
```

```
from sklearn.preprocessing import LabelEncoder
from scipy.stats import kstest
data=pd.read_excel("CO2.xlsx")
data=data.drop(['Unnamed: 0'],axis=1)
le = LabelEncoder()
standard_scaler  = preprocessing.MinMaxScaler()
#特征工程第一步：数据预处理
#针对离散变量的标签
catagories=['隔夹层位置','沉积韵律','产水段长度','产水段位置','水平段位置']
for cat in catagories:
    t=le.fit_transform(data[cat].astype(str))
    data[cat]=t
#这一个属性有些特殊
dis=[]
for items in data['平行井距离/m']:
    if items=='无平行井':
        dis.append(-1)
    else:
        dis.append(items)
data['平行井距离/m']=dis
#数值变量的归一化
numbers=['地层倾角','油层有效厚度/m','非均质（洛伦兹系数）','投产含水/f','产水段非均质程度（渗透率倍数）',\
         '平行井采液速度/rm3/d','吞吐时机/f','周期注气量/sm3','注气速度/sm3/d','关键时间/d',\
             '开井后采液速度/rm3/d','CO2 吞吐增油量']
max_tar=data['CO2 吞吐增油量'].max()
min_tar=data['CO2 吞吐增油量'].min()
#for num in numbers:
 #   data[num]=(data[num]-data[num].min())/(data[num].max()-data[num].min())

#用决策树回归进行测试
from sklearn import tree
cart=tree.DecisionTreeRegressor()
X=data[['吞吐时机/f','平行井采液速度/rm3/d','平行井距离/m','周期注气量/sm3','非均质（洛伦兹系数）']]
Y=data['CO2 吞吐增油量']
from sklearn.metrics import r2_score
dtree=tree.DecisionTreeRegressor(max_depth=5)
dtree.fit(X,Y)
y_predict=dtree.predict(X)
print(r2_score(y_predict,Y))
#用最简单的多层感知机也就是神经网络测试
from sklearn.neural_network import MLPRegressor
NNmodel = MLPRegressor([100,50,10],learning_rate_init= 0.001,activation='relu',\
     solver='adam', alpha=0.1,max_iter=300000)  # 神经网络
#Second 训练数据
print('start train!')
```

```
NNmodel.fit(X,Y)
print('end train!')
#Third 检验训练集的准确性
y_predict_1=NNmodel.predict(X)
print(r2_score(y_predict_1,Y))
#很显然多层感知机萎了
#尝试可视化决策树
dot_data=tree.export_graphviz(dtree,
                out_file=None,
                feature_names=['t','v','d','q','l'],
                filled=True,
                rounded=True
                )
graph=pydotplus.graph_from_dot_data(dot_data)
display(Image(graph.create_png()))
df=pd.DataFrame(np.c_[np.r_[Y],np.r_[y_predict]])
df.columns=['real','predicted']
#sns.boxplot( data=df['real'], width=0.5, linewidth=1.0, palette="Set3",whis=30)
sns.boxplot( data=df, width=0.5, linewidth=1.0, palette="Set3",whis=10)
plt.plot(Y-y_predict)
plt.xlabel("sample")
plt.ylabel("bias")
```