机器学习在水力压裂预测中应用程序使用说明

一．总述

中国非常规资源丰富，致密气产量占 1/3，致密油快速发展。现有压裂设计方法难以均匀物性分段，亟需运用深度学习，识别表征储层可压性分布，指导压裂设计优化。前人针对人工智能预测储层物性及参数反演进行了相关实验。但仍缺乏运用机器学习训练钻头破岩数据与破裂压力的关系，建立数据模型。

本程序是基于机器学习建立的水力压裂预测的应用程序，其功能是通过筛选钻头破岩参数、TVD 等基础数据，利用其机器学习模型，计算并与破裂压力数据拟合，实现对水裂压力的预测并构建相关的残差图与回归图。

其功能包含了数据导入，数据缺失值处理，数据标准化，回归器和图像输出等。

二．数据导入功能

使用 pandas 的数据导入功能，在模块的函数种通过输入数据文件的路径将需要处理的文件导入到程序中。并将数据按照给定比例分成 test 组和 train 组，同时通过相关系数函数生成输入数据的相关系数图。**图 1.1** 为本指导中导入的数据，共 6 种特征，分别为井深，钻头尺寸，扭矩，机械转速，钻压和钻速。标签值为破裂压力。

	A	B	C	D	E	F	G
1	破裂压力伪随机数据						
2	井深m	钻头尺寸mm	扭矩KN·m	机械钻速m/h	钻压KN	转速r/min	破裂压力Mpa
3	3253	165.1	4.83	1.338132868	70	50	44.18
4	3272	165.1	4.25	1.129532983	60	50	42.78
5	3357	165.1	4.8	0.11815076	30	40	42.61
6	3363	165.1	5.67	0.608214311	80	50	43.24
7	3373	165.1	8.95	0.468678216	20	30	46.31
8	3394	165.1	5.32	1.123971809	50	40	42.75
9	3412	165.1	6.18	1.117446498	80	50	44.19
10	3452	165.1	5.31	1.401803901	50	50	45.96
11	3461	165.1	7.81	0.334882285	30	40	46.87
12	3463	165.1	8.34	1.50597036	20	50	46.4
13	3465	165.1	6.81	1.511230065	60	50	48.47
14	3480	165.1	6.08	0.744924635	60	40	47.71
15	3482	165.1	6.21	0.402282026	60	50	46.95

图 1.1 导入数据示例

三．数据缺失值处理

在数据处理过程中，数据值缺失是数据分析中经常遇到的问题之一，程序中包含了三种可以选择的对于数据缺失值的处理方法。

四．数据标准化功能

数据的标准化（normalization）是将数据按比例缩放，使之落入一个小的特定区间。数据经过标准化不仅可以提升模型的收敛速度还可以提高模型的精度，在程序中包含可以选择的三种数据标准化方法。

五．回归器

回归器的回归器包含经典的 SVM，GaussianProcessRegressor 等回归器以及基于集成学习思想的 RandomForestRegressor，BaggingRegressor 等总计 21 种回归器。

全部 21 种回归器如下所示：

{'KernelRidge', 'SVR',

'SGDRegressor', 'KNeighborsRegressor',

'GaussianProcessRegressor','DecisionTreeRegressor',

'RandomForestRegressor','ExtraTreesRegressor',

'ELMRegressor','GenELMRegressor',

'GradientBoostingRegressor','MLPRegressor',

'KernelRegression','RVR','KernelSGD',

'AdaBoostRegressor','BaggingRegressor','StackingRegressor',

'VotingRegressor','HistGradientBoostingRegressor','LSSVR'}

在进行回归预测时提供多种选择，方便找到结果较优的回归器。**图 4.1** 展示了 main 函数中 4 种回归器的调用方法，其余回归器调用方法与此相同。

```
elif opt=='RVR':
    reg = RVR(kernel='rbf')
    modelname=RVR
elif opt=='KernelSGD':
    reg = KernelSGD(n_iter=10)
    modelname=KernelSGD
elif opt=='None':
    pass
stime = time.time()
reg.fit(X_train, y_train)
print("Time for "+opt+" fitting: %.3f" % (time.time() - stime))
return modelname,reg
```

图 4.1 回归器调用举例

六．图像输出

1. 相关系数图

相关系数图展示了各特征之间的相关系数，方便进行数据分析，图 5.1.1 为代码，图 5.1.2 为相关系数图举例。

```
correlations = X.corr()
c_map = plt.cm.get_cmap('coolwarm')
fig, ax = plt.subplots(figsize=(10,10))
sns.heatmap(correlations, vmax=1.0, center=0, fmt='.2f',
                square=True, linewidths=.5, cmap=c_map, annot=True,
                cbar_kws={"shrink": .70})
classes = ('Well depth','Bit size','Torque',
           'Drilling speed','Drilling weight','Speed','Rupture pressure')
ax.set_xticklabels(classes)
ax.set_yticklabels(classes)
plt.savefig('Correlation Matrix.pdf',dpi=600,bbox_inches='tight')
plt.show();
```

图 5.1.1 相关系数图 python 代码

图 5.1.2 相关系数图

2. 残差图

残差图程序（residualdraw.py）是基于 yellowbrick 模块修改的残差图的生成函数，它分别展示了训练集和测试集的 R-Square 的值以及图像，在'main'中被执行调用。

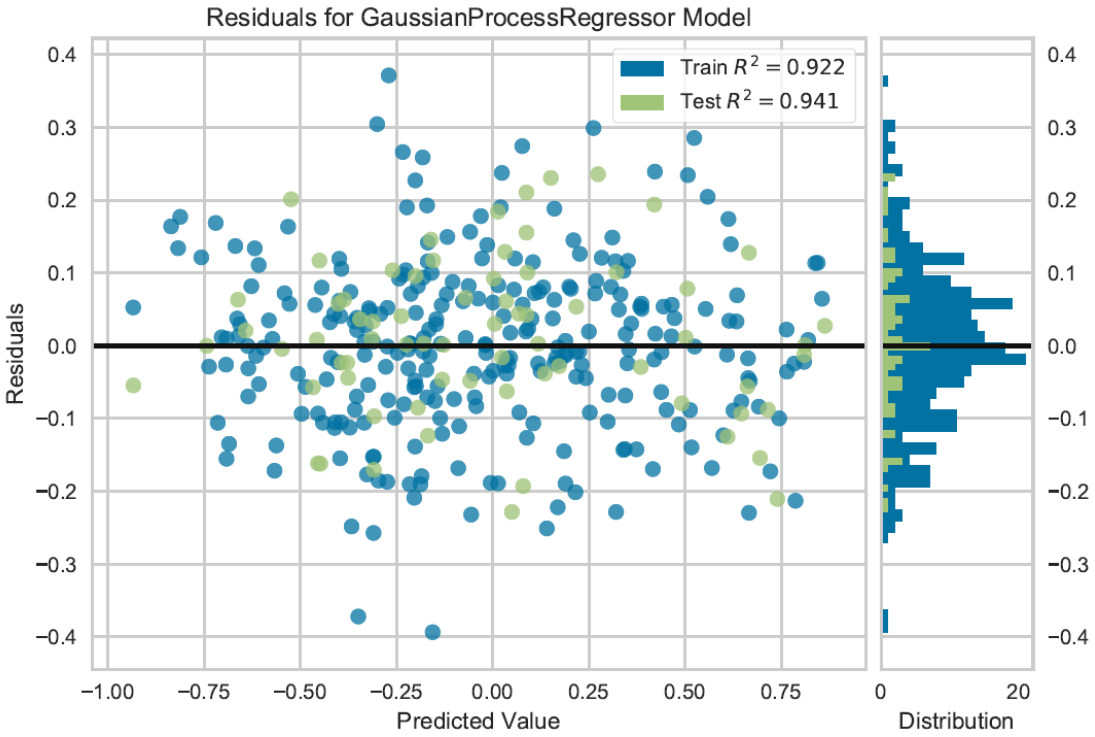

图 5.2.1 高斯进程回归器残差图

残差图生成选中回归器所产生的真实值与预测值之间的残差，蓝色为训练集，绿色为测试集。

3. 回归图

回归图直观的展示了预测值和真实值之间的关系，图 5.3.1 为高斯进程回归器的回归图。

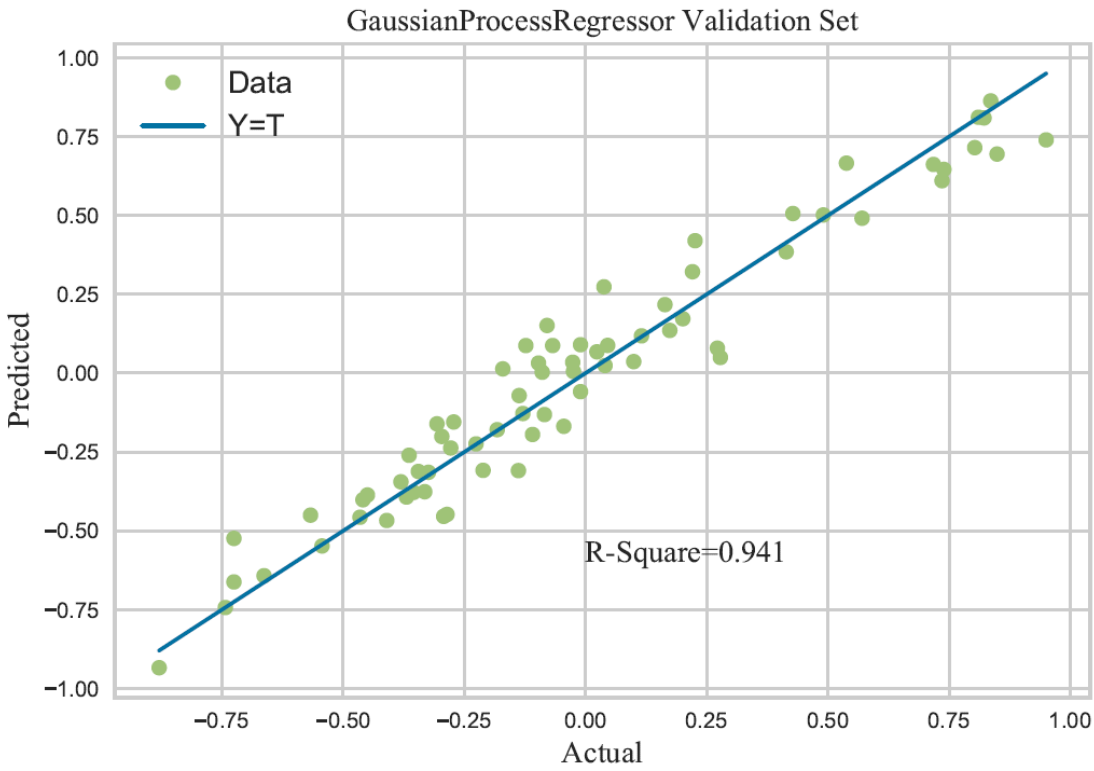

图 5.3.1 高斯进程回归图

七. 窗口界面

窗口界面如**图 6.1** 所示，在 Data Preprocessing Method 一栏可以选择 SimpleImputer， IterativeImputer，KNNImputer 和 MissingIndicator 四种方法在 Standardization Method 一栏可以选择 StandardScaler, MinMaxScaler, Normalizer 三种方法。

在 Rgeressor Selection 可以选择使用回归器的种类

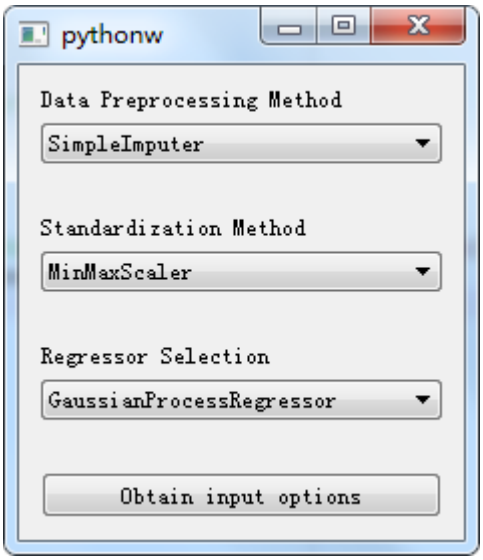

图 6.1 交互界面

全部选择后点击 Obtain input options 即可执行程序，得到输入数据的相关系数图，所选择回归器对应的残差图和回归图以及 RMSE、R-Square、Pearson 这三个计算结果。

RMSE 作为评价真实值与预测值之间的差异，是误差指标，理论上 RMSE 的值越接近于 0 越好。

R-Square 以及 Pearson Coefficient 为分数指标，作为评价回归模型的准则之一，越接近于 1 表示模型越好。

八．库的调用

SQWidget

可以实现窗口界面的调用，使程序交互性更强。

import pandas as pd

Pandas 是一个开源的、有 BSD 开源协议的库，它为 Python 编程语言提供了高性能、易于使用的数据架构以及数据分析工具。它提供了被称为 DataFrame 和 Series 的数据抽象，通过管理索引来快速访问数据、执行分析和转换运算，甚至可以绘图（用 matplotlib 后端）。

matplotlib.pyplot 和 seaborn

python 中两个有关作图的库，支持多种图片产生以及处理方式，可以使结果表达更清晰，直观。

回归器的介绍：

KNeighborsRegressor：基于 k 最近邻的回归，通过对训练集中最近相邻的目标进行局部插值来预测目标。主要参数有 n_neighbors ，weights， leaf_size 和 algorithm 等。

DecisionTreeRegressor：决策树是将所有的数据都落到叶子节点，既可以做分类也可以做回归。主要参数 max_features ：None（所有），log2，sqrt，特征小于 50 的时候一般使用所有的。max_depth ：数据少或者特征少的时候可以不管这个值，如果模型样本量多，特征也多的情况下，可以尝试限制。min_samples_split：如果某节点的样本数少于 min_samples_split，则不会继续再尝试选择最优特征来进行划分如果样本量不大，不需要管这个值。如果样本量数量级非常大，则推荐增大这个值。min_samples_leaf ：这个值限制了叶子节点最少的样本数，如果某叶子节点数目小于样本数，则会和兄弟节点一起被剪枝，如果样本量不大，不需要管这个值。

SVR：SVR 是支持向量回归(support vector regression)的英文缩写，是支持向量机(SVM)的重要的应用分支。SVR 回归，就是找到一个回归平面，让一个集合的所有数据到该平面的距离最近。主要参数有 kernel，degree，gamma 等。

RandomForestRegressor：随机森林是 Bagging 的一个拓展变体，是集成学习的一个分支，因为它依靠于决策树的集成。随机森林是一种元估计量，它适合数据集各个子样本上的许多分类决策树，并使用平均数来提高预测准确性和控制过度拟合。

如果 bootstrap = True（默认值），则使用 max_samples 参数控制子样本大小，否则将使用整个数据集构建每棵树。主要参数有 n_estimators，max_depth，min_samples_split，min_samples_leaf。

Kernel Ridge Regression：即使用核技巧的岭回归（L2 正则线性回归），它的学习形式和 SVR（support vector regression）相同，但是两者的损失函数不同：KRR 使用的 L2 正则均方误差；SVR 使用的是带 L2 正则的ϵ-insensitive loss。KRR 有近似形式的解，并且在中度规模的数据时及其有效率，由于 KRR 没有参数稀疏化的性能，因此速度上要慢于 SVR（它的损失函数有利于得到稀疏化的解）。主要参数有 kernel，alpha，gamma，degree 等。

GaussianProcessRegressor：高斯过程回归（Gaussian Process Regression, GPR）是使用高斯过程（Gaussian Process, GP）先验对数据进行回归分析的非参数模型（non-parameteric model）。主要参数有 alpha，kernel 和 optimizer 等。

ExtraTreesRegressor：该算法与随机森林算法十分相似，都是由许多决策树构成。但该算法与随机森林有两点主要的区别：

1、随机森林应用的是 Bagging 模型，而 ET 是使用所有的训练样本得到每棵决策树，也就是每棵决策树应用的是相同的全部训练样本；

2、随机森林是在一个随机子集内得到最佳分叉属性，而 ET 是完全随机的得到分叉值，从而实现对决策树进行分叉的。主要参数有：n_estimators，criterion，max_depth，min_samples_split，min_samples_leaf 等。

ELMRegressor & GenELMRegressor ELM 是一种新型的快速学习算法，对于单隐层神经网络，ELM 可以随机初始化输入权重和偏置并得到相应的输出权重。

RVR：在数学中，关联向量机（RVM）是一种机器学习技术，它使用贝叶斯推理来获得用于回归和概率分类的简约解。RVM 具有与支持向量机相同的功能形式，但提供了概率分类。

SGDRegressor：SGD 代表随机梯度下降（Stochastic Gradient Descent:）：每次估计每个样本的损失梯度，并随着强度进度表（即学习率）的降低而更新模型。主要参数有 loss，penalty，alpha 等。

GradientBoostingRegressor：GB 以渐进的阶段方式建立加性模型； 它允许优化任意微分损失函数。 在每个阶段，将回归树拟合到给定损失函数的负梯度上。主要参数有 loss，learning_rate，n_estimators，subsample 等。

KernelRegression：通过自动带宽选择来实现 Nadaraya-Watson 内核回归。其输入参数为 kernel 和 gamma 值，在程序中通过

MLPregressor：全称为多层感知器回归器。该模型使用 LBFGS 或随机梯度下降来优化平方损耗。输入参数有 hidden_layer_sizes，activation，solver，alpha，batch_size,learning_rate 等。

KernelSGDRegressor： SDGRegressor 融合了 Kernel 影响因素后的回归器模块，器输入参数有 kernal 和 SDGRegressor 的原有参数。

AdaBoostRegressor：一种元估计器，它首先将回归器拟合到原始数据集上，然后将回归器的其他副本拟合到同一数据集上，但根据当前预测的误差调整实例的权重。其输入参数为 base_estimator，n_estimators，learning_rate，loss 等。

BaggingRegressor：一个集合元估计器，它使每个基本回归器都适合原始数据集的随机子集，然后将其单个预测（通过投票或平均）进行汇总以形成最终预测。 通过将随机化引入其构造过程中，然后对其进行集成，通常可以将这种元估计器用作减少黑盒估计器（例如，决策树）的方差的方法。其输入参数有 base_estimator，n_estimators 等。

RidgeCV：带有内置交叉验证的回归器。 默认情况下，它执行通用交叉验证，这是一种有效的“留一法”交叉验证的形式。其输入参数为 alphas，fit_intercept，normalize，scoring 等。

LinearSVR：类似于带有参数 kernel ='linear'的 SVR，但它是用 liblinear 而不是 libsvm 来实现的，因此它在选择罚分和损失函数时具有更大的灵活性，并且应更好地扩展到大量样本。 此类同时支持密集和稀疏输入。其输入参数为 epsilon，tol，C，loss 等。

StackingRegressor：一种 estimator 到最终回归器的堆栈。 堆叠概括包括堆叠各个估算器的输出，并使用回归器计算最终预测。 堆叠允许通过将每个单独的估算器的输出用作最终估算器的输入来利用其强度。 estimators_拟合在完整的 X 上，而 final_estimator_是使用 cross_val_predict 使用基础估计量的交叉验证预测来训练的。其输入参数有 estimators，final_estimator，cv，n_jobs 等。

LinearRegression：普通最小二乘线性回归器。 LinearRegression 使用系数 w =（w1，…，wp）拟合线性模型，以最小化数据集中观察到的目标与通过线性近似预测的目标之间的平方余数。其输入参数为 fit_intercept，normalize，copy_X，

n_jobs 等。

VotingRegressor：适用于不适合的估算器的预测投票回归器。 表决回归器是一个集合元估计器，它适合多个基本回归器，每个基本回归器都位于整个数据集中。 然后，将各个预测取平均以形成最终预测。其输入参数为 VotingRegressor，weights，n_jobs 等。

enable_hist_gradient_boosting：启用基于直方图的梯度增强估计器。 这些估计器的 API 和结果可能会发生变化，而不会发生任何弃用周期。 动态导入此文件会、设置为 ensemble 模块的属性。

HistGradientBoostingRegressor：一种基于直方图的梯度增强回归树。 对于大型数据集（n_samples>=10000），此估计器比 GradientBoostingRegressor 快得多。此估算器具有对缺失值（NaNs）的本地支持。 在训练过程中，树木种植者会根据潜在收益，在每个分割点上学习是否应将缺失值的样本分配给左孩子或右孩子。 进行预测时，因此将具有缺失值的样本分配给左或右子级。 如果在训练过程中没有遇到给定特征的缺失值，则将具有缺失值的样本映射到样本数量最多的那个孩子。其输入参数为 loss，learning_rate 等。