回归器集合程序使用说明

一. 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。

破裂压力伪随机数据						
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
3253	165.1	4.83	1.338132868	70	50	44.18
3272	165.1	4.25	1.129532983	60	50	42.78
3357	165.1	4.8	0.11815076	30	40	42.61
3363	165.1	5.67	0.608214311	80	50	43.24
3373	165.1	8.95	0.468678216	20	30	46.31
3394	165.1	5.32	1.123971809	50	40	42.75
3412	165.1	6.18	1.117446498	80	50	44.19
3452	165.1	5.31	1.401803901	50	50	45.96
3461	165.1	7.81	0.334882285	30	40	46.87
3463	165.1	8.34	1.50597036	20	50	46.4
3465	165.1	6.81	1.511230065	60	50	48.47
3480	165.1	6.08	0.744924635	60	40	47.71
3482	165.1	6.21	0.402282026	60	50	46.95

图 1.1 导入数据示例

二. 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。

1. Simpleimputer

Simpleimputer 类提供了输入缺失值的基本策略。缺失值可以用常量值或使用缺失值所在列的统计信息（平均值、中位数或最频繁）进行填充。

调用格式为 Imputer = SimpleImputer(missing_values=nan, strategy='mean')

参数缺失值 missing_values 索引为 nan，填充策略 strategy 选取平均值 mean 作为填充方法。

2. IterativeImputer

通过以循环方式将具有缺失值的每个要素建模为其他要素的函数来估算缺失值的策略。

调用格式为 Imputer = IterativeImputer(max_iter=10,

initial_strategy='constant',random_state=0)

参数最大插补回合数为 10，以 constant 常量作为初始化策略，并锁定随机状态。

3.KNNImputer

每个样本的缺失值都是使用在训练集中找到的 n 个最近邻居的平均值估算的。

调用格式为 KNNImputer(n_neighbors=10, weights="uniform")

最邻近值数量 n_neighbors 定义为 10 个，将权重 weights 定义为所有临近值拥有相同权重

4.MissingIndicator

是一种提供缺失值的二进制指示器。

调用格式格式为 transformer =

ColumnTransformer(transformers=[('vanilla_features', SimpleImputer

(strategy='constant', fill_value=-1), list(range(X.shape[1]))),

('indicate_features', MissingIndicator(error_on_new=False), [])],

remainder='passthrough')

三. 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。

1. StandardScaler

通过去除均值并缩放到单位方差来标准化特征

样本 x 的标准分数计算如下：

z = (x-u) / s

其中 u 是训练样本的平均值，如果 with_mean = False，则为零，而 s 是训练样本的标准偏差，如果 with_std = False，则为 1。

通过计算训练集中样本的相关统计信息，对每个特征进行独立居中和缩放。然后将平均值和标准偏差存储起来，以便通过变换在以后的数据上使用。

数据集的标准化是许多机器学习估计器的普遍要求：如果各个特征看起来或多或少

不像标准正态分布数据(例如均值和单位方差为 0 的高斯),它们可能会表现不佳。

调用格式为 ss = StandardScaler()

2.MinMaxScaler

通过将每个要素缩放到给定范围来变换要素。

该估计器分别缩放和转换每个特征，以使其处于训练集的给定范围内，例如在 0 和 1 之间。

转换方式为：

X_std = (X-X.min（轴= 0)) / (X.max（轴= 0）-X.min（轴= 0))

X_scaled = X_std *（最大-最小）+最小

最小值，最大值= feature_range。

此变换通常用作零均值，单位方差缩放的替代方法。

调用格式为 ss = MinMaxScaler(copy=True,feature_range=(-1, 1))

参数 feature_range 定义了数据转换后的格式为列矩阵。

3.Normalizer

将样本分别归一化为单位范数。

有至少一个非零分量的每个样本（即数据矩阵的每一行）都独立于其他样本进行重新缩放，以使其范数（l1，l2 或 inf）等于 1。

该转换器可以同时处理密集的 numpy 数组和 scipy.sparse 矩阵。

调用格式为 ss = Normalizer()

四．回归器

回归器的回归器包含 sklearn 中经典的 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

关于回归器的更多支持请参阅

https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

https://scikit-learn.org/stable/modules/classes.html?highlight=ensemble#module-sklearn.ensemble

五. 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。

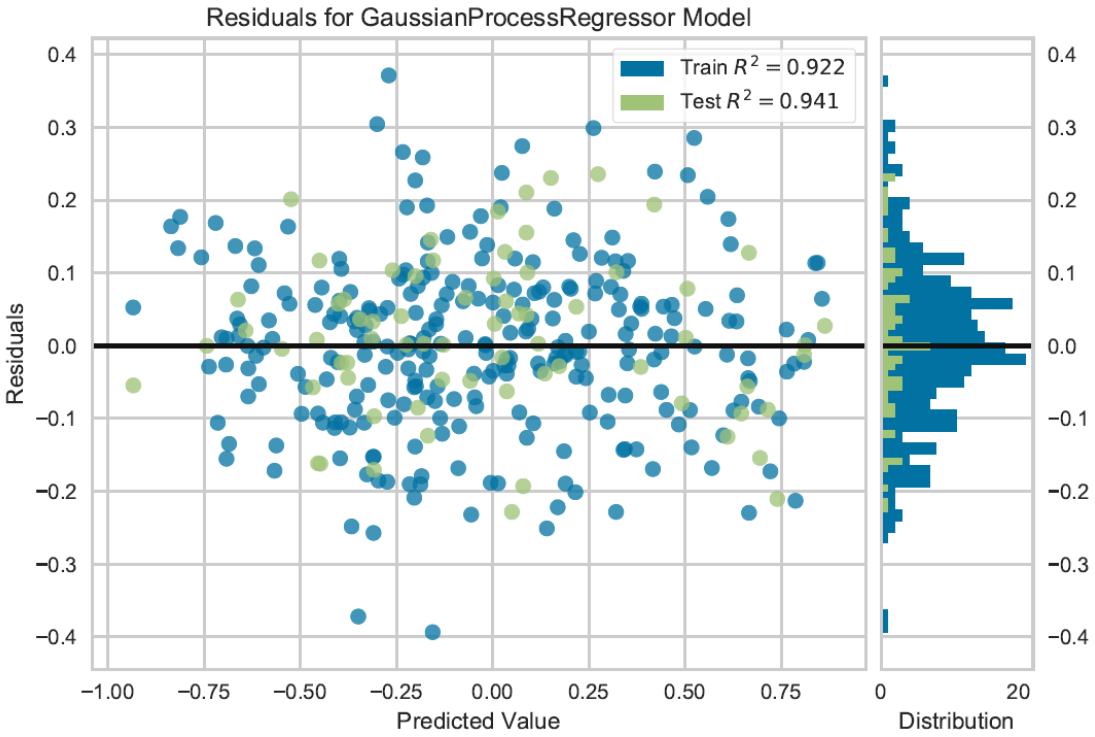

图 5.2.1 高斯进程回归器残差图

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

关于残差图的更多支持请参阅

https://www.scikit-yb.org/en/latest/api/regressor/residuals.html

致谢残差图模块原作者

Author: Rebecca Bilbro

Author: Benjamin Bengfort

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。

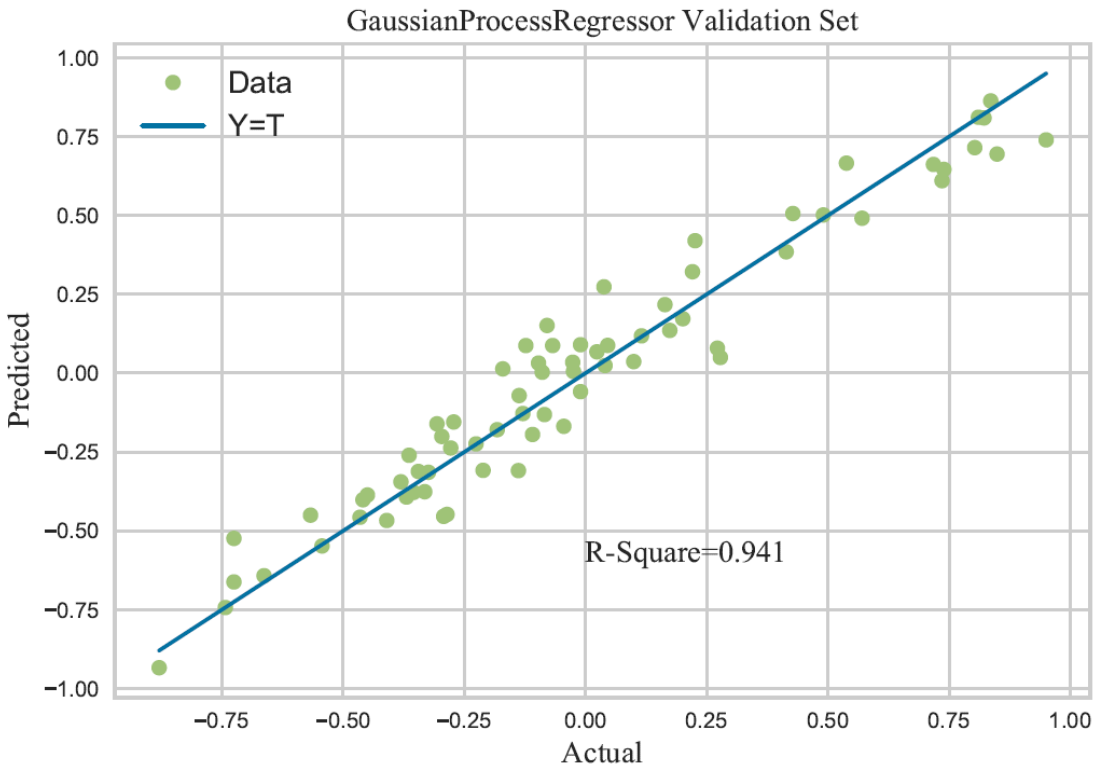

图 5.3.1 高斯进程回归图

六. 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

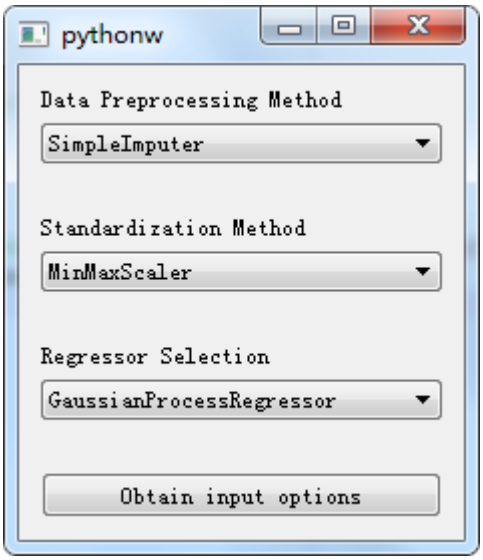

图 6.1 交互界面

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。