**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。



**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。

**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')

**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。

**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()

**五．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',

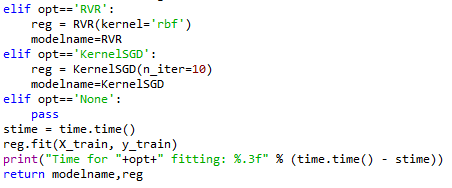
'GradientBoostingRegressor','MLPRegressor',

'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。



**图4.1** 回归器调用举例

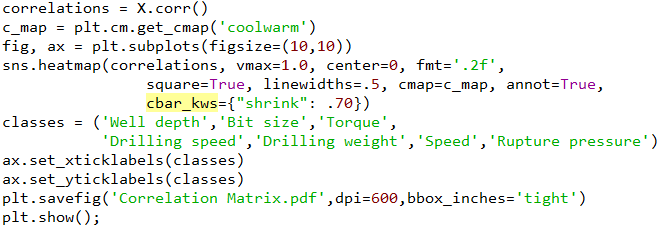
关于回归器的更多支持请参阅

<https://scikit-learn.org/stable/supervised_learning.html#supervised-learning> <https://scikit-learn.org/stable/modules/classes.html?highlight=ensemble#module-sklearn.ensemble>

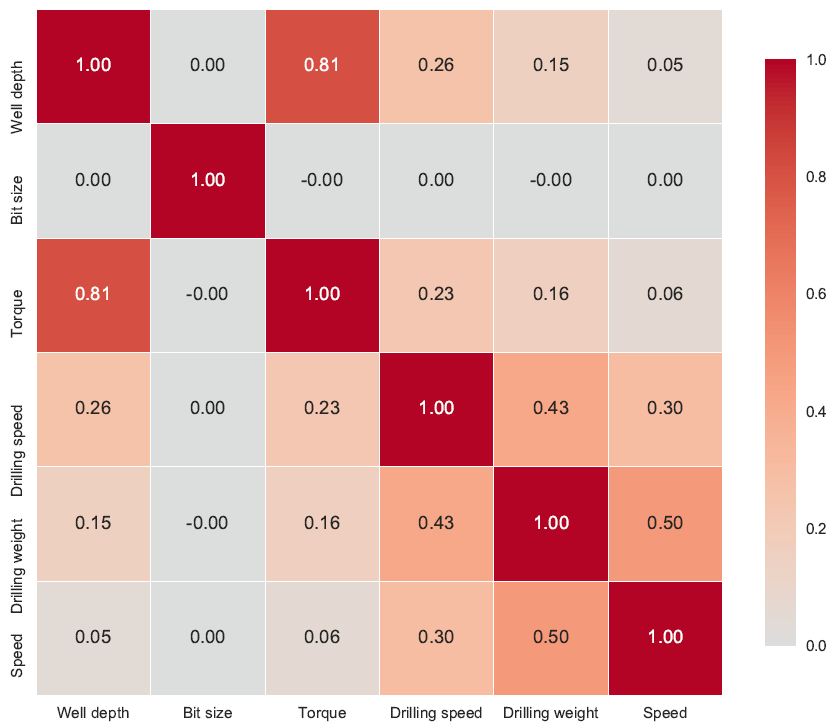
**六．图像输出**

**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。

****

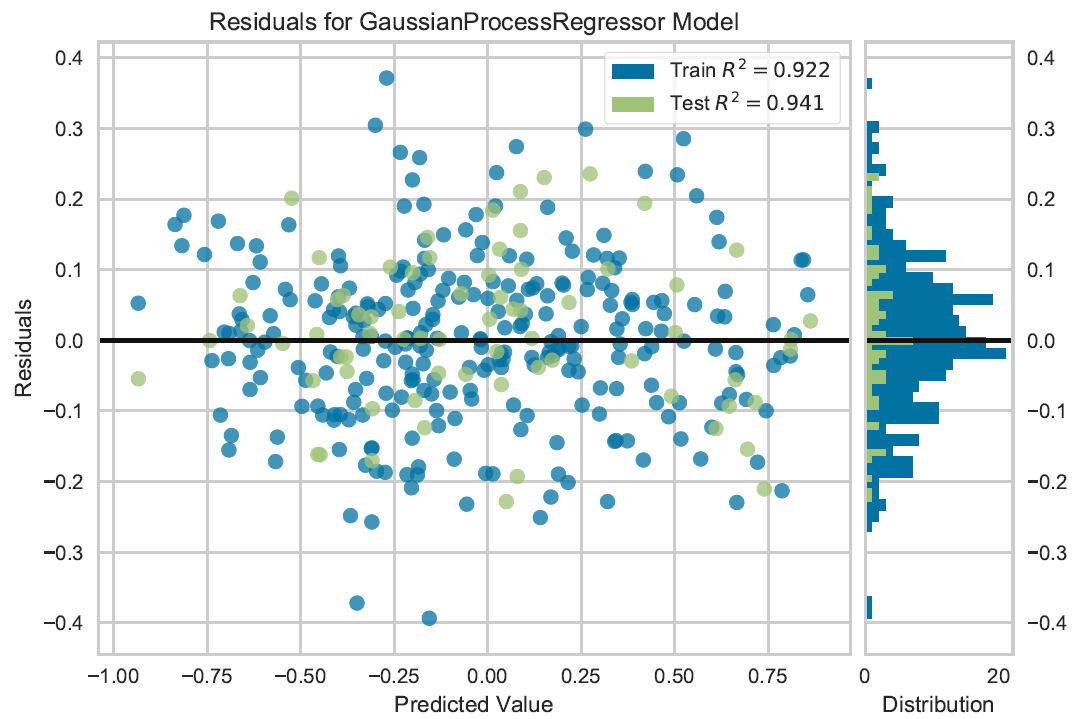
**图5.1.1** 相关系数图python代码

****

**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

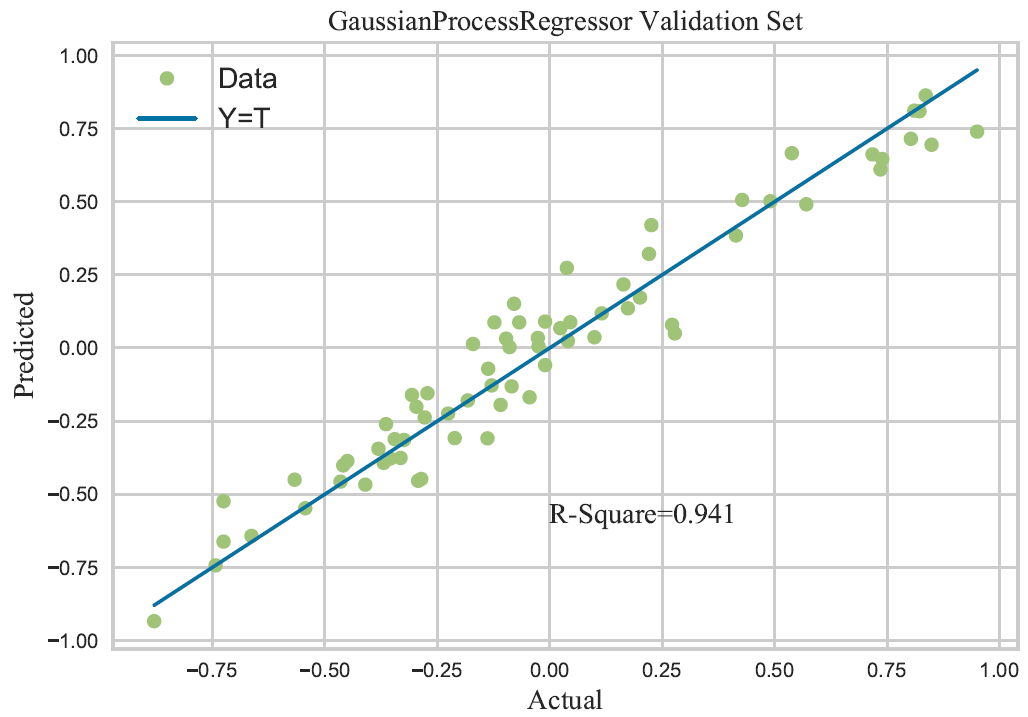
致谢残差图模块原作者：

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。



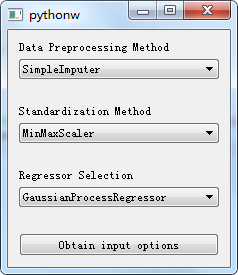
**图 5.3.1** 高斯进程回归图

**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。

**from residualdraw import ResidualsPlot**

调用残差图函数在主程序中执行，可以生成回归的残差图，图内包含测试集与训练集，可以对回归做更加直观的分析。

**numpy和 np.random.seed(1234)**

NumPy(Numerical Python) 是 Python 语言的一个扩展程序库，支持大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。随机种子的设定是为了使程序具有复现性。

**import time**

调用时间模块，可以计算出不同回归器拟合以及预测所花费的时间，也是评价回归器的指标之一

**train\_test\_split**

train\_test\_split函数用于将矩阵随机划分为训练子集和测试子集，并返回划分好的训练集测试集样本和训练集测试集标签。

train\_data：被划分的样本特征集

train\_target：被划分的样本标签

test\_size：如果是浮点数，在0-1之间，表示样本占比；如果是整数的话就是样本的数量

random\_state：是随机数的种子。

**SimpleImputer,IterativeImputer，KNNImputer, MissingIndicator**

**和enable\_iterative\_imputer**

以上四种插补方法都是用来以不同的方式填补数据的缺失值。

以SimpleImputer为例，指令：SimpleImputer(missing\_values=nan,strategy='mean',fill\_value=None ,copy=True，add\_indicator=False)

missing\_values：代表缺失值，一般情况下就是空值，也就是np.nan。

strategy:也就是你采取什么样的策略去填充空值，总共有4种选择。分别是mean，median，most\_frequent以及constant。如果是mean，则该列则由该列的均值填充。而median则是中位数，most\_frequent则是众数。需要注意的是，如果是constant,则可以将空值填充为自定义的值，这就要涉及到后面一个参数了，也就是fill\_value。如果strategy='constant',则填充fill\_value的值。

copy:则表示对原来没有填充的数据的拷贝。

add\_indicator:如果该参数为True，则会在数据后面加入n列由0和1构成的同样大小的数据，0表示所在位置非空，1表示所在位置为空。相当于一种判断是否为空的索引。

关于其余的插补方法以及更详细的解释请查阅<https://scikit-learn.org/stable/modules/classes.html#module-sklearn.impute>

**MinMaxScaler, StandardScaler,Normalizer：**

以上三种为经典的数据预处理方法。

正则化Normalizer:

正则化的过程是将每个样本缩放到单位范数（每个样本的范数为1），如果后面要使用如二次型（点积）或者其它核方法计算两个样本之间的相似性这个方法会很有用。

标准化/无量纲化Standardscaler:

将数据按比例缩放，使之落入一个小的特定区间内，标准化后的数据可正可负，一般绝对值不会太大。该方法要求数据近似为高斯分布，标准化后数据均值为0，方差为1。

区间缩放MinMaxScaler:

将属性缩放到一个指定的最大和最小值（通常是1-0）之间，这可以通过preprocessing.MinMaxScaler类实现。

**GridSearchCV**

详尽搜索指定参数的估计值。 重要的参数都可以用来预测。 GridSearchCV实现“拟合”和“得分”方法。 如果在所使用的估计器中实现了“预测”，“预测\_proba”，“决策功能”，“变换”和“逆变换”，则还可以实现它们。 通过对参数网格进行交叉验证的网格搜索，优化了用于应用这些方法的估计器的参数。

r2\_score，r2（确定系数）回归得分函数。 最佳可能得分为1.0，并且可能为负（因为该模型可能会更差）。 不管输入特征如何，始终预测y的期望值的恒定模型将获得0.0的r2分数。

**mean\_squared\_error**

均方误差回归误差模块。

**mean\_absolute\_error**

平均绝对误差回归误差模块。

**Pearsonr**

计算皮尔逊相关系数和p值以测试非相关性。 皮尔逊相关系数测量两个数据集之间的线性关系。 严格来说，Pearson的相关性要求每个数据集都必须呈正态分布。 像其他相关系数一样，该系数在-1和+1之间变化，其中0表示无相关。 -1或+1的相关性暗示精确的线性关系。 正相关表明，随着x的增加，y也随之增加。 负相关性表示随着x增加，y减小。 p值大致表示不相关的系统产生的数据集具有Pearson相关性至少与从这些数据集计算得出的极端一样高的概率。 p值并不完全可靠，但对于大于500左右的数据集可能是合理的。输入参数为参数X,Y。

**sp\_randint**

是均匀离散随机变量生成模块。 离散随机变量是从标准形式定义的，可能需要一些形状参数才能完成其规格说明。 可以将任何可选的关键字参数传递给RV对象的方法

**StratifiedKFold**

一种分层的K折交叉验证器。 提供训练/测试索引以将数据拆分为训练/测试集。 此交叉验证对象是K-Fold的变体，它返回分层的折痕。 折叠是通过保留每个类别的样品百分比来进行的。其输入参数为n\_splits，shuffle，random\_state。

**SVC**

C-supoort向量的分类器。 该实现基于libsvm。 拟合时间至少与样本数量成平方比例，超过成千上万的样本可能不切实际。 对于大型数据集，请考虑使用sklearn.svm.LinearSVC或sklearn.linear\_model.SGDClassifier代替，可能在sklearn.kernel\_approximation.Nystroem转换器之后。 多类支持根据一对一方案进行处理。其输入参数为C，kernel，degree，gamma等。

**GaussianProcessClassifier**

是基于拉普拉斯近似的高斯过程分类器（GPC）。 该实现基于Rasmussen和Williams的高斯机器学习过程（GPML）的算法3.1、3.2和5.1。 在内部，拉普拉斯近似用于通过高斯近似非高斯后验。 当前，该实现仅限于使用逻辑链接功能。 对于多类别分类，安装了几个二进制的单反其余分类器。 注意，该类因此未实现真正的多类拉普拉斯近似。其输入参数为kernel，optimizor等。

**RationalQuadratic**

内核可以被视为具有不同特征长度尺度的RBF内核的尺度混合（无穷大）。 它由长度比例参数和比例混合参数进行参数化。 目前仅支持length\_scale为标量的各向同性变体。其输入参数为length\_scale，alpha，length\_scale\_bounds等。

**load\_iris**

可以用来加载并返回虹膜数据集（分类）。 虹膜数据集是经典且非常容易的多类别分类数据集。其输入参数为return\_X\_y，as\_frame，返回值为data。

**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](https://baike.baidu.com/item/%E9%AB%98%E6%96%AF%E8%BF%87%E7%A8%8B/4535435" \t "_blank)（Gaussian Process, GP）先验对数据进行回归分析的[非参数模型](https://baike.baidu.com/item/%E9%9D%9E%E5%8F%82%E6%95%B0%E6%A8%A1%E5%9E%8B/2010527" \t "_blank)（non-parameteric model）。主要参数有alpha，kernel和optimizer等。

**ExtraTreesRegressor：**该[算法](http://lib.csdn.net/base/datastructure)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：**通过与sklearn兼容的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。

**KernelSGDRegresso**r： 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。 动态导入此文件会将sklearn.ensemble.HistGradientBoostingClassifier sklearn.ensemble.HistGradientBoostingRegressor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。