机器学习个人补充报告

—　lab01 回归问题

学　　院　电子与信息工程学院

专　　业　计算机科学与技术

授课老师　李　洁

学　　号　1853790

姓　　名　庄镇华

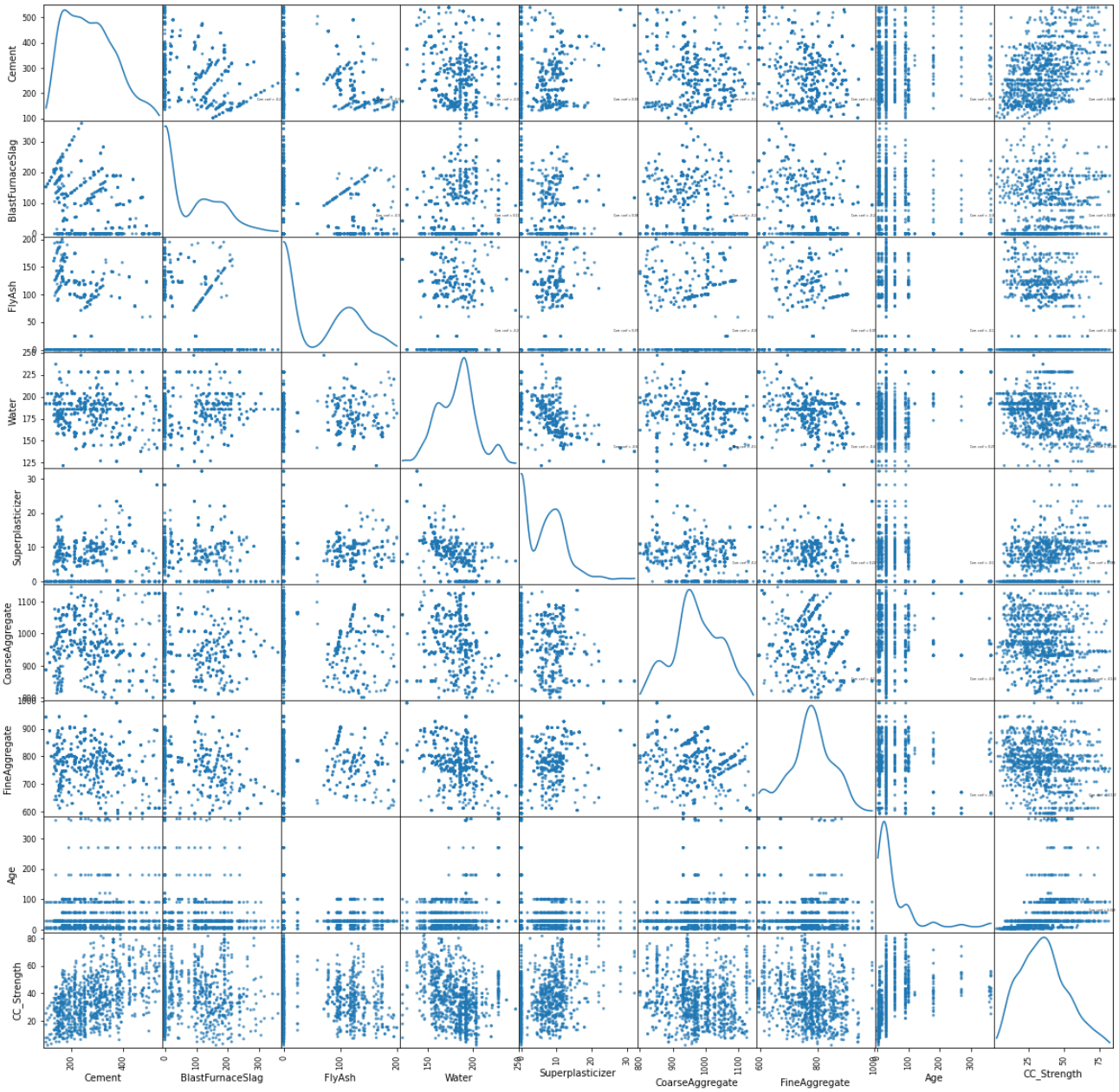
完成日期　2021.06.20

说明：原报告中主要使用了线性回归、支持向量机回归与高斯过程回归等模型，并且对模型进行了 K 折交叉验证，最后手写了线性回归模型。本次补充报告使用了更多的 11 个模型，并对它们进行了对比分析。此外，还使用特征选择和网格搜索进行了模型优化。

一、数据分析

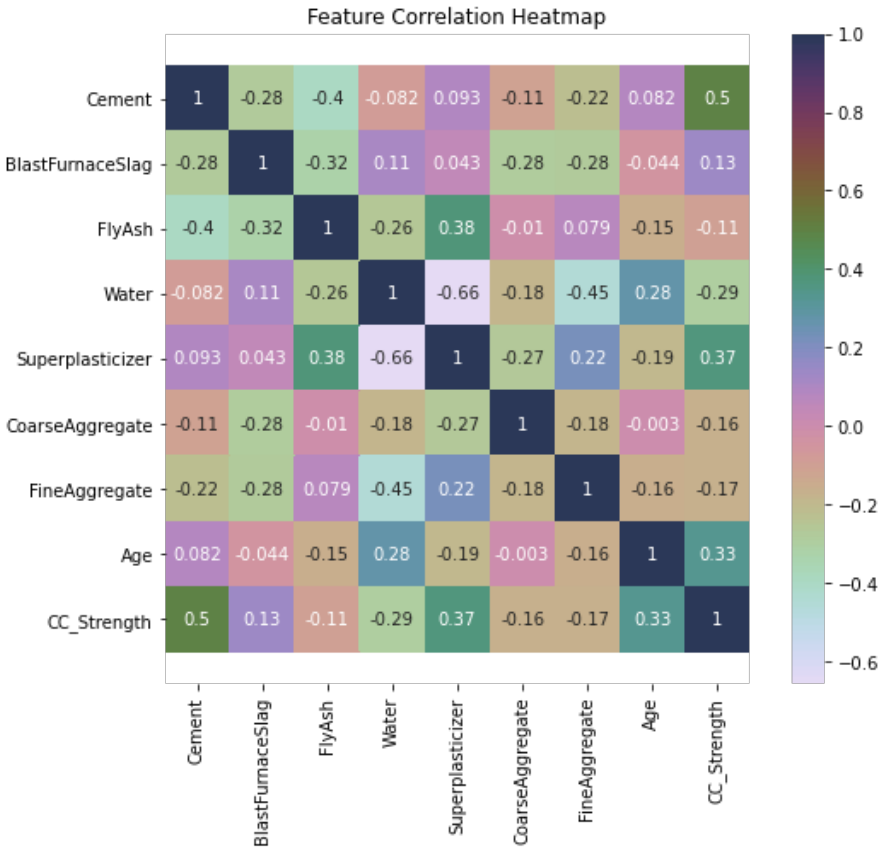
1.1 计算特征间的相关程度

绘制原始数据各个输入特征之间的散点密度图。可以通过对角线部分的散点图直观查看各个特征两两之间的关系，通过对角线部分的密度图查看各个变量的密度分布。



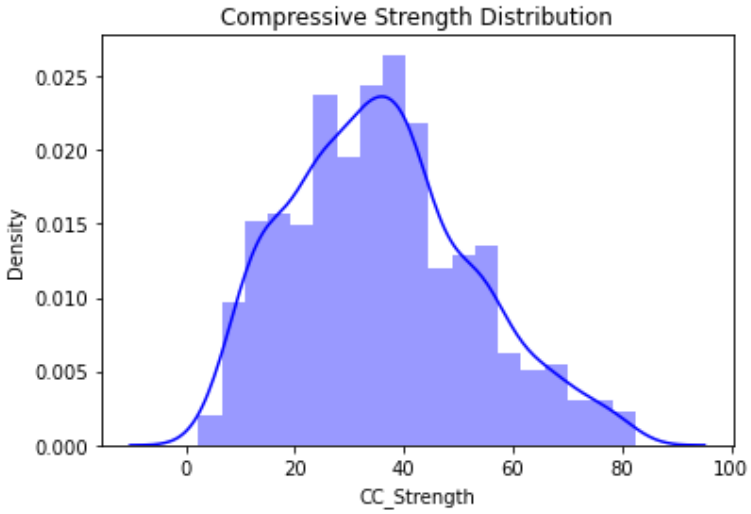
计算原始数据各个输入特征之间的相关程度，观察图像可以得出与因变量 CC_Strength 相关程度较高的有 Cement、Superplasticizer 和 Age。

并且两两特征之间相关程度较高的有 Water 和 Superplasticizer，Water 和 FineAggregate 特征。



1.2 观察因变量的大致分布

利用柱状图和曲线图相结合，查看因变量 CC_Strength 的大致分布。



二、模型训练

主要采用了 11 种回归模型，包括线性回归、最小角回归、岭回归、支持向量机回归、K 最近邻回归、决策树回归、随机森林回归、Adaboost 回归、GBRT 回归、Bagging 回归和极端随机树回归。

下面简要介绍这些模型的数学原理、优缺点及搭建过程。

2.1 线性回归

概述

线性回归模型的数学形式为

$$Y = X\beta + \varepsilon$$

其中 Y 为因变量，X 为自变量， β 为回归系数， ε 为随机误差项。线性规划的目标是寻求一个合适的回归系数 β，使得误差最小。

当 X 为列满秩矩阵时，回归系数可由普通的最小二乘估计方法求得

$$\hat{\beta}_{OLS} = arg\min_{\beta \in R^d} |Y - X\beta|^2 = (X^T X)^{-1} X^T Y$$

特点

线性回归的缺点是对数据要求较高，若 X 不是列满秩矩阵，则不能使用最小二乘法求解回归系数。

2.2 最小角回归

概述

当矩阵 X 不满足列满秩时，不能采用普通最小二乘法来求解回归系数，此时引入惩罚方法。最小角回归方法是在普通线性模型中增加 L 惩罚项，对于普通线性模型，Lasso 估计为

$$\hat{\beta}_{Lasso} = \arg\min_{\beta \in R^d} (|Y - X\beta|^2 + \lambda \sum_{j=1}^{d} |\beta_j|)$$

特点

可将一部分系数压缩到 0，降低 X 的维度，达到减小模型复杂度的目的。

2.3 岭回归

概述

当协变量间相互独立时，采用普通最小二乘法估计参数 β 具有很好的性质。但是当协变量维度增加时，难免会存在相关关系，此时设计矩阵 X 将不再满足列满秩，我们称这样的设计矩阵为病态的。

当 X 呈病态时，$X^T X$接近奇异，此时$\hat{\beta}_{OLS}$虽然具有最小方差，但是$\hat{\beta}_{OLS}$值很大，导致精度比较差，表现相当不稳定。针对设计矩阵为病态的情况，众多学者提出了不同的改进方法，用得较为广泛的是岭回归。对于线性模型，岭回归的数学形式为

$$\hat{\beta}_{Ridge} = arg\min_{\beta \in R^d} \left(|Y - X\beta|^2 + \lambda \sum_{j=1}^{d} |\beta_j|^2 \right) = \frac{1}{1+\gamma} \widehat{\beta_{oLS}}$$

特点

岭估计是对普通最小二乘估计以同一比例进行压缩，但是不会将原本非零的系数压缩至 0。

2.4 支持向量机

概述

SVM 的学习策略是间隔最大化，可形式化为一个求解凸二次规划的问题，也等价于正则化的合页损失函数的最小化问题。SVM 的学习算法就是求解凸二次规划的最优化算法。SVM 回归的目标函数为

$$\min_{w,b} C\sum_{i=1}^{N} E_{\epsilon}(\hat{y_i} - y_i) + \frac{|w|^2}{2}$$

特点

SVM 学习问题可以表示为凸优化问题，因此可以利用已知的有效算法发现目标函数的全局最小值。但 SVM 算法对大规模训练样本难以实施。

2.5 KNN

概述

K 最近邻算法又称为 KNN 算法。所谓的 K 最近邻，就是指最接近的 K 个邻居（数据），即每个样本都可以由它的 K 个邻居来表达。kNN 算法的核心思想是，在一个含未知样本的空间，可以根据离这个样本最邻近的 k 个样本的数据类型来确定样本的数据类型。

特点

KNN 模型训练时间快，但对不相关的功能和数据规模敏感。

2.6 决策树回归

概述

决策树的输入训练数据是一组带有类别标记的样本，构造结果是一棵多叉树。树的分支节点一般表示为一个逻辑判断。某一节点上选择特征的标准是在该节点上选取能对该节点处的训练数据进行最优划分的属性。划分的标准是信息增益，即划分前后数据集的熵的差异。取能带来最大信息增益的那个特征作为当前划分标准。

信息熵

$$Info(D) = -\sum_{i=1}^{m} p_i \, log_2(p_i)$$

信息增益

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times info(D_j)$$

2.7 随机森林回归

概述

随机森林属于 Bagging 类算法，而 Bagging 方法又属于集成学习一种方法，集成学习的大致思路是训练多个弱模型打包起来组成一个强模型，强模型的性能要比单个弱模型好很多，其中的弱模型可以是决策树、SVM 等模型，在随机森林中，弱模型选用决策树。

在训练阶段，随机森林采样从输入训练数据集中采集多个不同的子训练数据集来依次训练多个不同决策树；在预测阶段，随机森林将内部多个决策树的预测结果取平均得到最终的结果。

特点

它的主要特点有具有极好的准确率；

能够有效地运行在大数据集上；能够处理具有高维特征的输入样本，而且不需要降维； 能够评估各个特征在分类问题上的重要性。

2.8 AdaBoost 回归

概述

Adaboost 是一种迭代算法，其核心思想是针对同一个训练集训练不同的分类器(弱分类器)，然后把这些弱分类器集合起来，构成一个更强的最终分类器（强分类器）。

特点

优点是可以将不同的分类算法作为弱分类器；相对于随机森林算法，Adaboost 充分考虑的每个分类器的权重。缺点是数据不平衡导致分类精度下降；训练比较耗时，每次重新选择当前分类器最优切分点。

2.9 GBRT 回归

概述

GBRT 是集成学习 Boosting 家族的成员，但是却和传统的 Adaboost 有很大的不同。Adaboost 利用前一轮迭代弱学习器的误差率来更新训练集的权重，这样一轮轮的迭代下去。GBRT 也是迭代，使用了前向分布算法，但是弱学习器限定了只能使用 CART 回归树模型，同时迭代思路和 Adaboost 也有所不同。

在 GBRT 的迭代中，假设前一轮迭代得到的强学习器是$f_{t-1}(x)$, 损失函数是$L(y, f_{t-1}(x))$， 则本轮迭代的目标是找到一个 CART 回归树模型的弱学习器$h_t(x)$，让本轮的损失函数$L(y, f_t(x)) = L(y, f_{t-1}(x) + h_t(x))$最小。

特点

它的优点主要有天然可处理不同类型的数据和对空间外的异常点处理非常健壮。

2.10 Bagging 回归

概述

Bagging 是由 Breiman 提出的一个简单的组合模型，它对原始数据集做很多次放回抽样，每次抽取和样本量同样多的观测值，放回抽样使得每次都有大约百分之三十多的观测值没有抽到，另一些观测值则会

重复抽到，如此得到很多不同的数据集，然后对于每个数据集建立一个决策树，因此产生大量决策树。对于回归来说，一个新的观测值通过如此多的决策树得到很多预测值， 最终结果为这些预测值的简单平均。

2.11 极端随机树回归

概述

极端随机树里的随机包含的意思是：特征随机、参数随机、模型随机、分裂随机，极端随机树算法与随机森林算法十分相似，都是由许多决策树构成。极限树与随机森林的主要区别：1、RandomForest 应用的是 Bagging 模型，ExtraTree 使用的所有的样本，只是特征是随机选取的，所以在某种程度上比随机森林得到的结果更加好 2、随机森林是在一个随机子集内得到最佳分叉属性，而极端随机树是完全随机的得到分叉值，从而实现对决策树进行分叉的。

2.12 模型搭建

主要利用 sklearn 库中模型进行搭建和训练。

第三方库引用

```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn.linear_model import Ridge
from sklearn import svm
from sklearn import neighbors
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import BaggingRegressor
from sklearn.tree import ExtraTreeRegressor
```

模型搭建

```
# 线性回归
lr = LinearRegression()
# 最小角回归
lasso = Lasso()
# 岭回归
ridge = Ridge()
# SVM 回归
SVM = svm.SVR()
# KNN 回归
KNN = neighbors.KNeighborsRegressor()
# Adaboost 回归
Ada = AdaBoostRegressor(n_estimators=100)#这里使用 100 个决策树
# 决策树回归
dtr = DecisionTreeRegressor()
# 随机森林回归
rfr = RandomForestRegressor(n_estimators=100)
# GBRT 回归
GBRT = GradientBoostingRegressor(n_estimators=100)
# Bagging
BR = BaggingRegressor()
```

```
# ExtraTree 极端随机树回归
```

模型训练

```
# 模型训练
lr.fit(X_train, y_train)
lasso.fit(X_train, y_train)
ridge.fit(X_train, y_train)
SVM.fit(X_train, y_train)
KNN.fit(X_train, y_train)
Ada.fit(X_train, y_train)
dtr.fit(X_train, y_train)
rfr.fit(X_train, y_train)
GBRT.fit(X_train, y_train)
BR.fit(X_train, y_train)
ETR.fit(X_train, y_train)
```

三、对比分析

3.1 评价指标

主要用 4 个指标来衡量不同及其学习方法的优劣，即 RMSE、MSE、MAE 和 R2 score。下列公式中，y_{test}（y）和$\hat{y}_{test}$（$\hat{y}$）分别表示真实测试数据和预测训练数据。

$$\frac{1}{m}\sum_{i=1}^{m}\left(y_{test}^{(i)}-\hat{y}_{test}^{(i)}\right)^2=MSE$$

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y_{test}^{(i)}-\hat{y}_{test}^{(i)}\right)^2}=\sqrt{MSE_{test}}=RMSE_{test}$$

$$\frac{1}{m}\sum_{i=1}^{m}\left|y_{test}^{(i)}-\hat{y}_{test}^{(i)}\right|=MAE$$

$$R^2=1-\frac{\sum_i\left(\hat{y}^{(i)}-y^{(i)}\right)^2}{\sum_i(\bar{y}-y^{(i)})^2}$$

3.2 对比分析

11 种模型的 4 个指标的实验结果如下表所示：

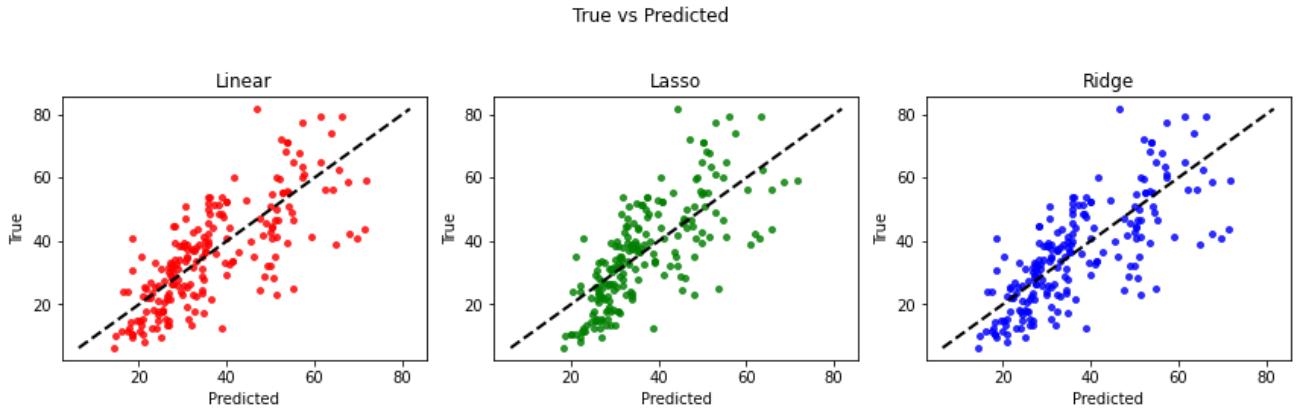
模型	RMSE	MSE	MAE	R2
LinearRegression	11.17	124.81	8.79	0.548376
LassoRegression	11.97	143.22	9.56	0.481768
RidgeRegression	11.17	124.85	8.80	0.548220
SVM-Regression	10.27	105.39	8.05	0.618657
KNN-Regression	9.55	91.14	7.52	0.670206
AdaBoost Regressor	8.04	64.68	6.66	0.765959
DecisionTreeRegressor	6.99	48.84	4.49	0.823284
RandomForestRegressor	5.33	28.40	3.76	0.897237

GBRTRegressor	5.35	28.61	3.88	0.896481
BaggingRegressor	5.53	30.55	3.88	0.889471
ExtraTreeRegressor	7.33	53.76	4.87	0.805477

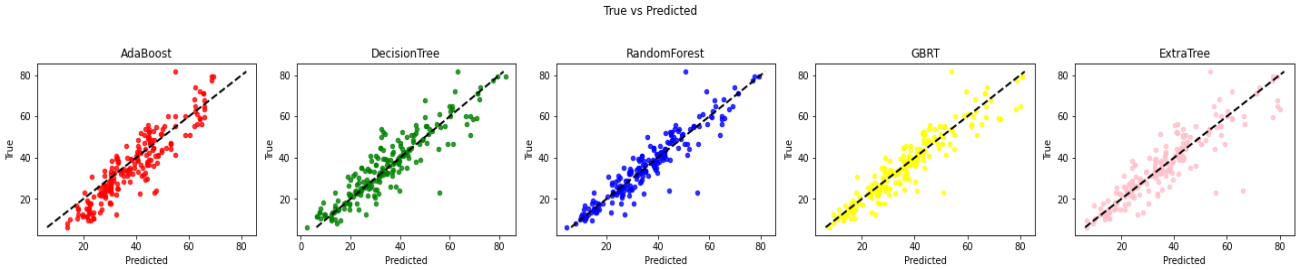
将预测结果与实际结果进行可视化对比，下图是 11 个线性模型的以预测值为横轴、真实值为纵轴的散点图。如果散点离对角线越近，则说明模型效果越好。

可以看出，和表格显示的结果相符合， RandomForestRegressor 模型的结果最符合实际值。

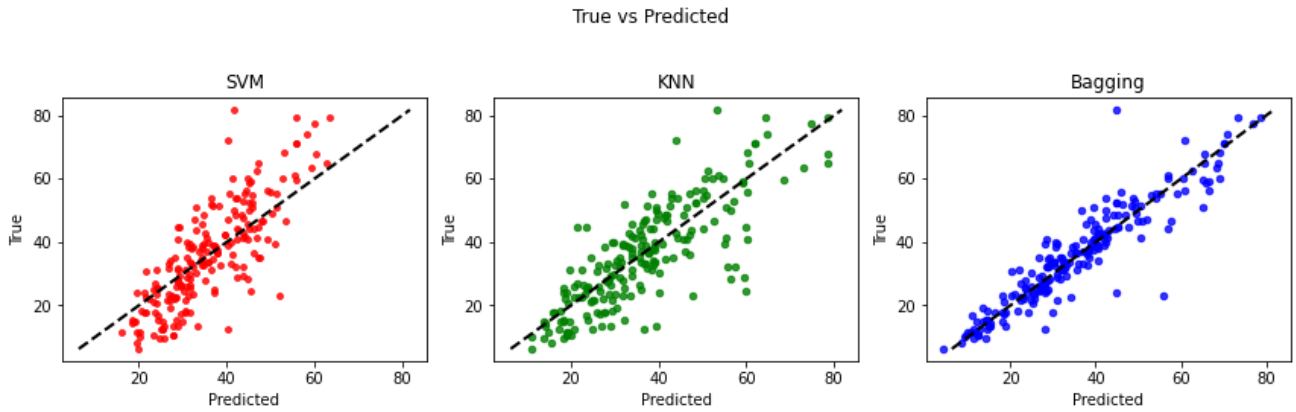
线性回归模型：



树类回归模型：



其他回归模型：



可以看出，RandomForestRegressor 模型的 4 种指标均达到最优，最适合本问题。关于原因分析，需要从随机森林的原理入手，简要列出如下：

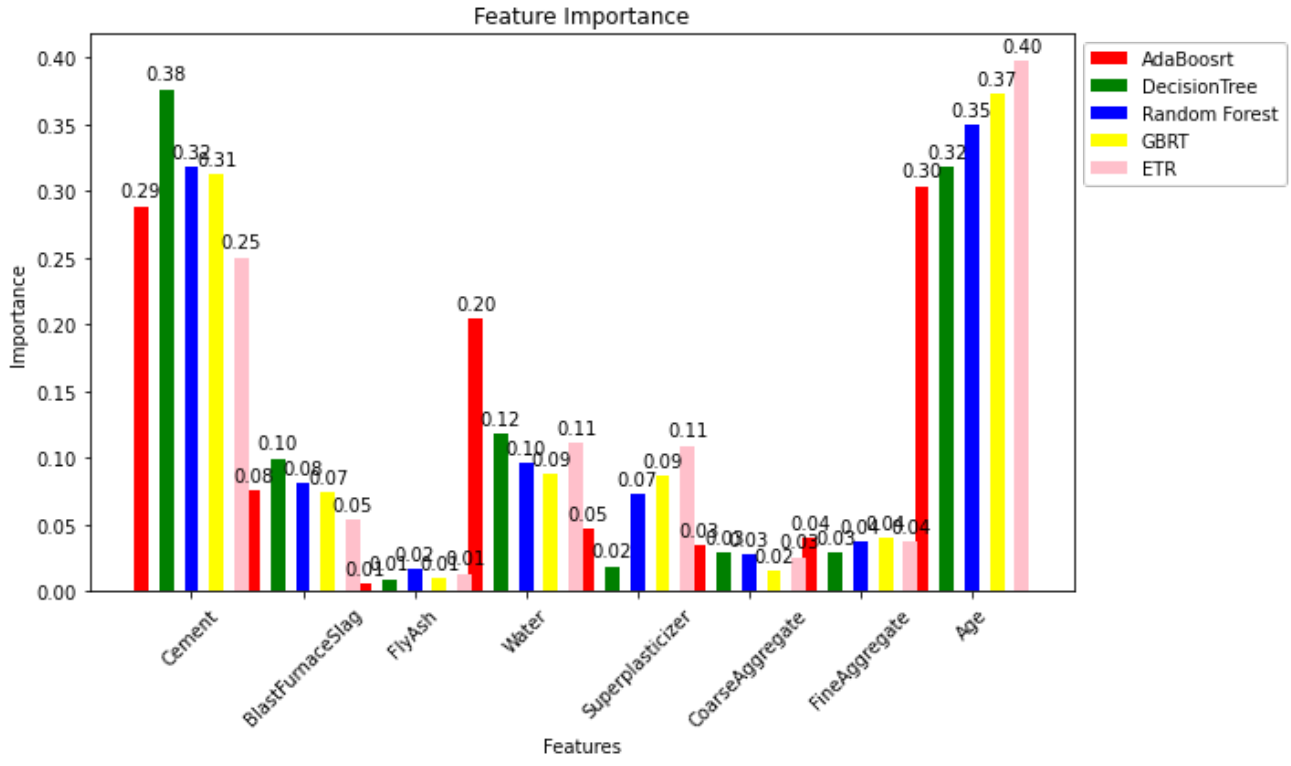
- 随机森林模型适合处理高维度的数据，并且由于其特征子集是随机选择的，因此不用做特征选择，这正适合没有做任何特征处理的数据
- 随机森林在创建的时候，使用的是无偏估计，因此模型泛化能力强
- 随机森林在训练过程中，能够检测到特征间的互相影响
- 随机森林即使在有很大一部分特征遗失的情况下，仍可以维持准确度

四、模型优化

4.1 特征选择

决策树类模型

在 sklearn 库中，决策树类别的回归算法可以生成 feature importance，用来表示不同特征对因变量的影响程度。下图是上文 11 个回归模型中的所有决策树类模型的 feature importance 对比柱状图。



可以看出，在预测混凝土强度时，Cement 和 Age 被视为最重要的特征，Flyash、Coarse 和 Fine 是最不重要的因素。下面去掉 FlyAsh 和 Fine 这两个特征，对其余特征进行选择，选用随机森林回归模型，得到实验结果如下：

	RMSE	MSE	MAE	R2
原始特征	5.21	27.24	3.67	0.896812
选择特征	5.19	26.90	3.60	0.900764

可以看到经过特征选择后，R2 指标升高，其余指标没有明显变化，这说明经过特征选择后模型鲁棒性更强，在测试集上精度更高。

线性回归模型

接下来用线性回归模型处理经过特征选取的数据。根据上文，我们可以得到，Fly Ash，Coarse Aggregate，Fine Aggregate 三个特征和 Concrete Compressive Strength 的相关性较差。

因此我们首先使用 sklearn 库函数选择 6 个最好的特征。

```
from sklearn.feature_selection import
SelectKBest,f_regression,chi2,f_classif,mutual_info_regression
# 选取最好的 6 个特征
x_new = SelectKBest(f_classif, k = 6).fit(X_train,
y_train).get_support(indices = True)
print(x_new)
```

然后进行测试，得到结果如下表所示：

	RMSE	MSE	MAE	R2
原始特征	9.68	93.84	7.70	0.657452
选择特征	9.66	93.56	7.71	0.658432

可以看到，减少了两个变量之后，模型的精确度几乎没有影响。R2 指标不减反增，这充分说明进行了特征选取后的线性回归模型比之前有更好的稳定性。

考虑到线性规划的核心原理，当一个变量对 y 的相关性较弱的时候，这个变量在这个式子中的价值也较小，如果过分关注这个变量可能会导致过拟合，在测试集上有不好的表现。因此当相关性较弱，此时删去这个变量，不但能够减少计算量，还有可能进一步提高模型的普适性、提高其在测试集上的精度。

4.2 网格搜索

原理

网格搜索是一项模型超参数（即需要预先优化设置而非通过训练得到的参数）优化技术，常用于优化三个或者更少数量的超参数，本质是一种穷举法。对于每个超参数，使用者选择一个较小的有限集去探索。然后，这些超参数笛卡尔乘积得到若干组超参数。网格搜索使用每组超参数训练模型，挑选验证集误差最小的超参数作为最好的超参数。网格搜索可以保证在指定的参数范围内找到精度最高的参数，但是这也是网格搜索的缺陷所在，要求遍历所有可能参数的组合，在面对大数据集和多参数的情况下，非常耗时。

特点

网格搜索适用于三四个（或者更少）的超参数（当超参数的数量增长时，网格搜索的计算复杂度会呈现指数增长，这时候则使用随机搜索），用户列出一个较小的超参数值域，这些超参数至于的笛卡尔积（排列组合）为一组超参数。网格搜索算法使用每组超参数训练模型并挑选验证集误差最小的超参数组合。

效果

对于普遍使用的线性回归模型，我们一般对其使用特征选择优化。而为了验证网格搜索的优化属性，我们选择了 K 最近邻回归模型进行验证。它的原理是：一个样本在特征空间中 k 个最相似（最邻近）的样本大多数属于同一个类别。使用它可以解决分类和回归问题。

我们选择的网格参数如下：

	weights	n_neighbors	p

第 1 类组合	uniform	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	/
第 2 类组合	distance	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	1, 2, 3, 4, 5

最终根据网格搜索得到的结果与原始结果对比如下：

	RMSE	**MSE**	**MAE**	**R2**
原始模型	8.15	66.38	6.45	0.76
网格搜索	7.58	57.40	5.63	0.79

最优参数如下表所示：

weights	**n_neighbors**	**p**
distance	7	2

由结果对比表中可得：不采用网格搜索的普通 K 近邻模型的 RMSE、MSE 和 MAE 误差明显高于采用网格搜索的 K 近邻模型，这说明使用网格搜索对包括 K 近邻在内的一些回归模型效果显著。