

**软件与微电子学院**

**数据挖掘期末作业：**

**官员晋升时间预测项目报告**

**组长：**

周彧祺

**组员：**

何金鸿

刘珏

吕美林

王木

目录

[1 数据获取 3](#_Toc43478549)

[2 数据预处理 4](#_Toc43478550)

[2.1 对个人介绍的处理 4](#_Toc43478551)

[2.2 对工作经历的处理 5](#_Toc43478552)

[2.3 获取升职经历 5](#_Toc43478553)

[3 模型 6](#_Toc43478554)

[3.1 数据统计 6](#_Toc43478555)

[3.2 相关性分析 9](#_Toc43478556)

[3.3 缺失值处理 12](#_Toc43478557)

[3.4 训练集和测试集生成 12](#_Toc43478558)

[3.5 模型简介 13](#_Toc43478559)

[3.6 模型评估 14](#_Toc43478560)

[3.7 模型训练 14](#_Toc43478561)

[3.7.1 Ridge 15](#_Toc43478562)

[3.7.2 Lasso 15](#_Toc43478563)

[3.7.3 RandomForestRegressor 16](#_Toc43478564)

[3.7.4 BayesianRidge 17](#_Toc43478565)

[3.7.5 SVR 18](#_Toc43478566)

[3.7.6 GradientBoostingRegressor 19](#_Toc43478567)

[3.7.7 ElasticNet 21](#_Toc43478568)

[3.7.8 SGDRegressor 22](#_Toc43478569)

[3.7.9 XGBRegressor 23](#_Toc43478570)

[3.7.10 ExtraTreesRegressor 26](#_Toc43478571)

[3.8 模型比较 28](#_Toc43478572)

[3.9 模型集成 28](#_Toc43478573)

[3.10 Stack模型集成 29](#_Toc43478574)

[4 项目总结 29](#_Toc43478575)

[5 小组分工 30](#_Toc43478576)

**官员晋升预测项目报告**

# 数据获取

我们使用python的requests包从<http://ldzl.people.com.cn/dfzlk/front/>爬取各省资料主页。

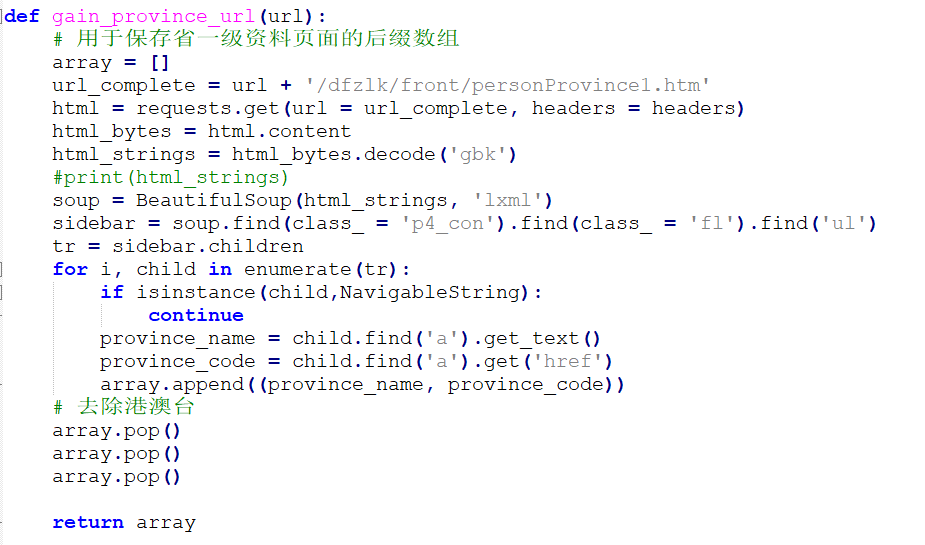


图 1 爬取省资料主页代码示例图

然后在省资料主页上爬取官员的个人主页信息。通过正则式匹配的方法，从省资料主页中筛选出党委班子、政府副职、市区县政领导的个人主页网址。最后从官员个人主页上获取官员的个人介绍和工作经历。

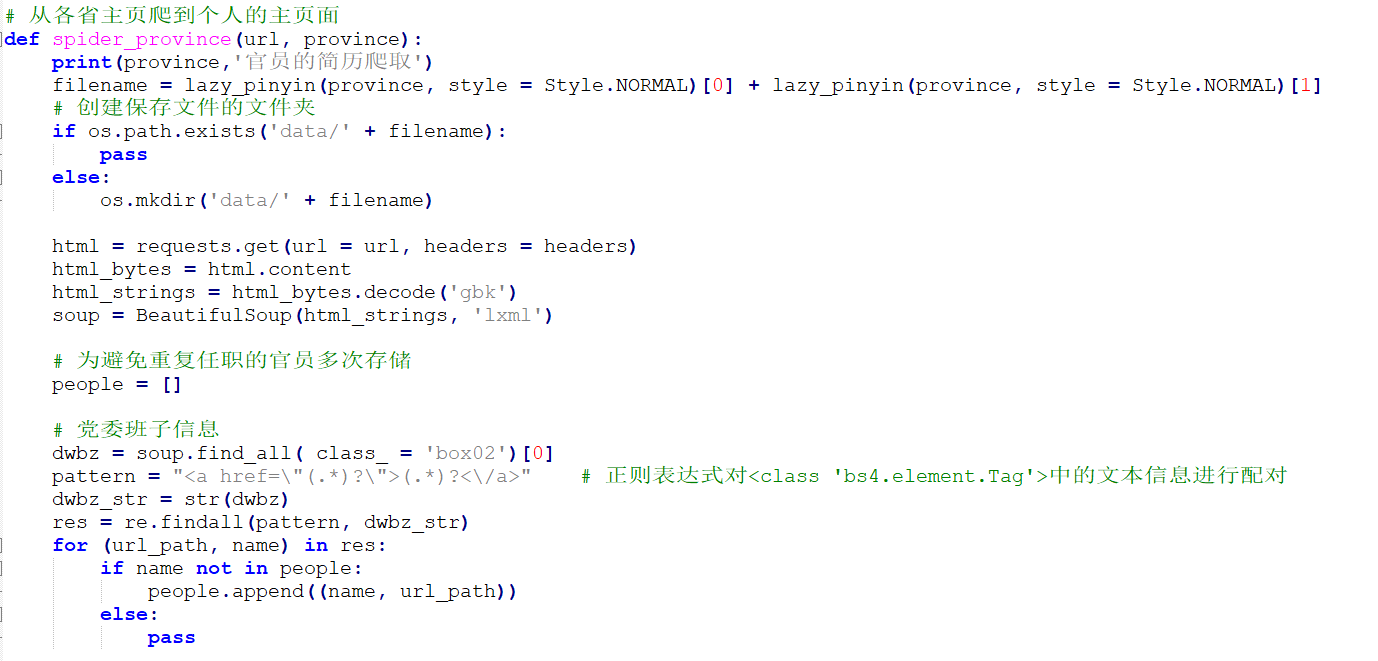


图 2 爬取个人主页代码示例图

爬取到的数据为json格式，分为个人介绍(introduction)和工作经历(experience)两个部分，示例如下图所示：

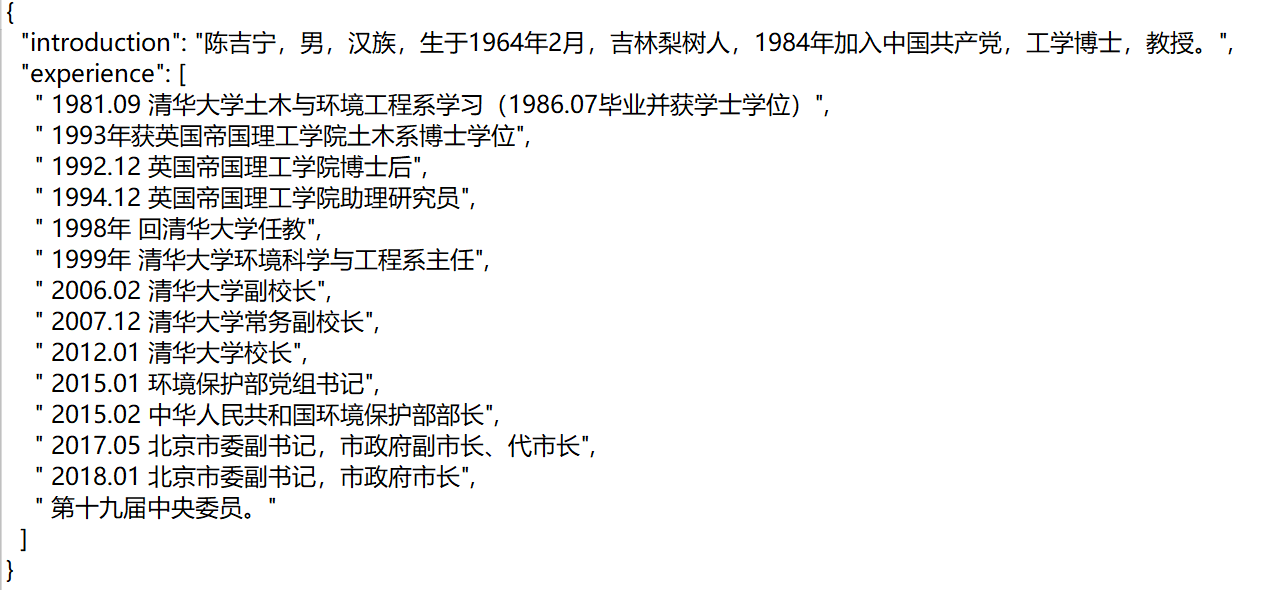


图 3 爬取的官员数据示例

# 数据预处理

## 对个人介绍的处理

在个人简介中我们获取的信息有姓名、性别、民族、出生年份、出生地、教育水平、工作年份、是否入党、入党年份。姓名、民族、出生地使用字符串存储。性别使用1表示男，0表示女。教育水平使用1表示本科，2表示硕士，3表示博士。是否入党使用1表示入党，0表示未入党。

## 对工作经历的处理

在对工作经历的处理中，我们提取入职时间和官职大小两个信息。官职被映射为按大小映射成0到13的数字，数字越大代表官职越大，映射规则如下：

* 我们建立直接映射官职和待定官职两个词表。直接映射官职可以直接映射成数字，待定官职的大小需要根据官职的修饰词来决定。

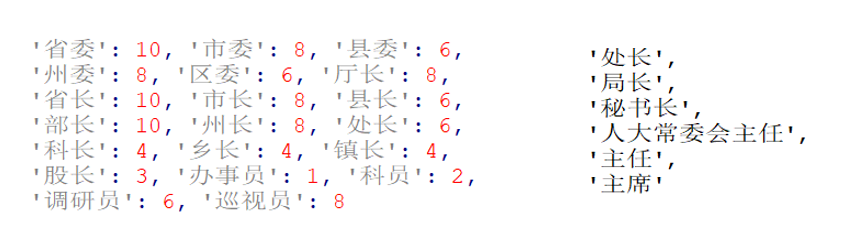


图 4 直接映射官职（左）和待定官职（右）

* 待定官职的大小需要通过地方修饰来判定。例如待定官职为处长，当地方修饰为8时官职大小为4，当没出现地方修饰时官职大小为6；待定官职为局长时，地方修饰为6时官职大小为4，地方修饰为8时官职大小为6。

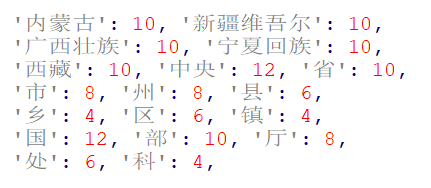


图 5 地方修饰词表

* 最后确定官职大小之前，若出现‘副’、‘助理’、‘常委’，则官职大小减一，若出现‘北京’、‘天津’、‘上海’、‘重庆’等直辖市，则官职大小加二。

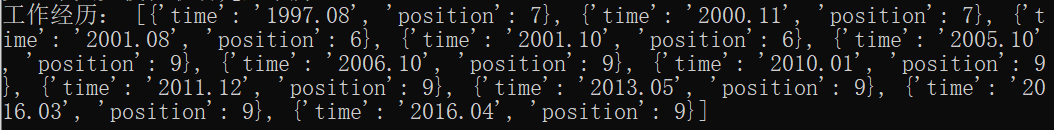


图 6 处理后的工作经历示例

## 获取升职经历

我们筛选出各个官员工作经历中的升职经历，将工作经历按照入职的年份从前到后排序，官职大小变大的两条工作经历即为升职经历。current\_post代表当前职位，next\_post代表升职后的职位，current\_post\_year代表开始担任原职位的时间，next\_post\_year代表升职时间。处理后共获得1494条升职经历。

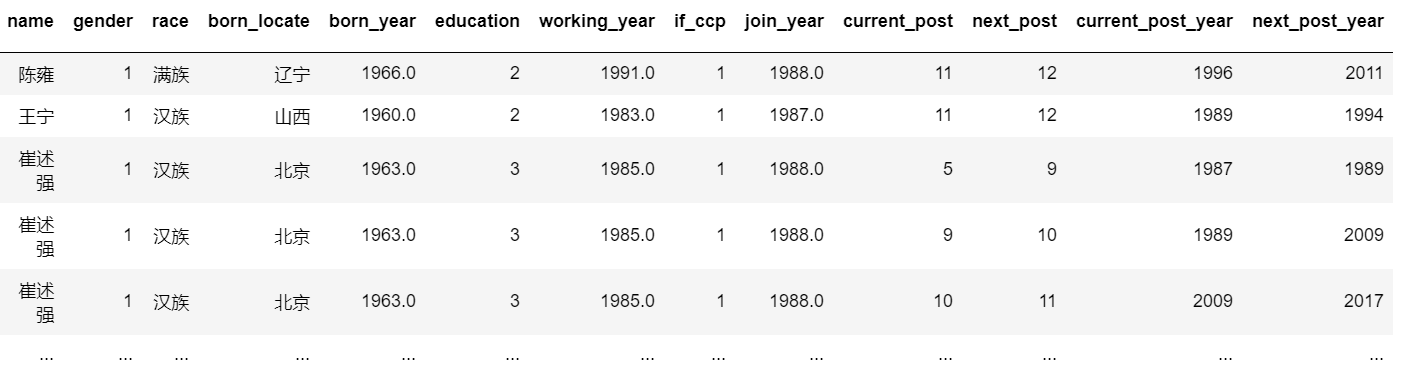


图 7 升职经历数据示例

# 模型

## 数据统计

首先我们对1494条升职经历中的性别进行统计，结果如下图所示。可以看出，我们的数据中男性的升职经历数量要远多于女性的。

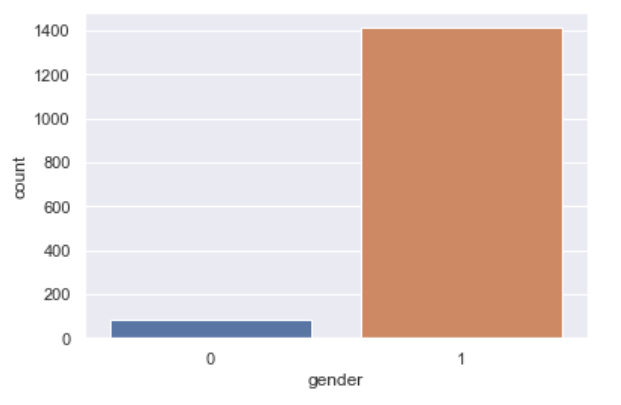


图 8 升职经历数据性别统计柱状图

对升职经历中的民族进行统计，结果如下图所示。可以看出数据中汉族的升职经历占大多数。

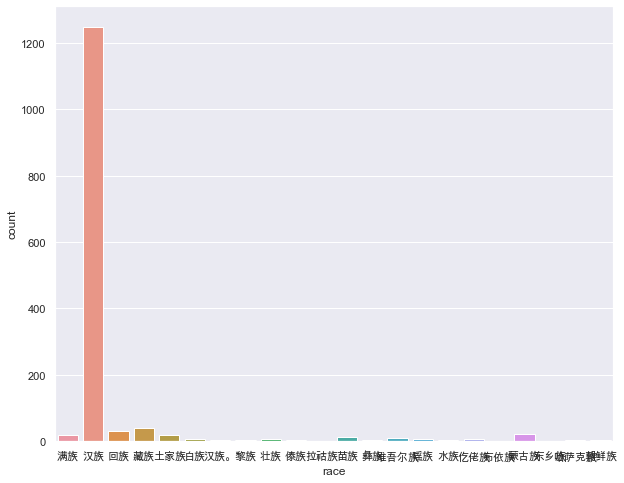


图 9 升职经历数据民族统计柱状图

对升职经历中的出生地进行统计，结果如下图所示。数据中出生地为山东、河南、湖北、重庆的人最多，个别省份如北京和海南的数据较少。

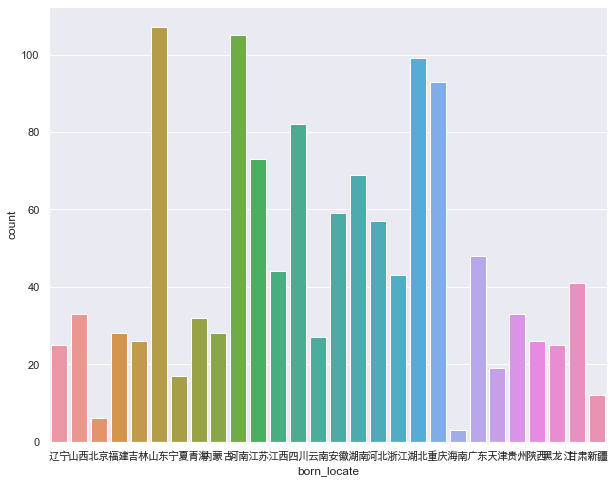


图 10 升职经历数据出生地统计柱状图

升职跨度为升职后职位大小减升职前职位大小的值（next\_post-current\_post）。对升职经历数据中的升职跨度进行统计，结果如下图所示。可以看出，升职跨度为1和2的数据占大多数。

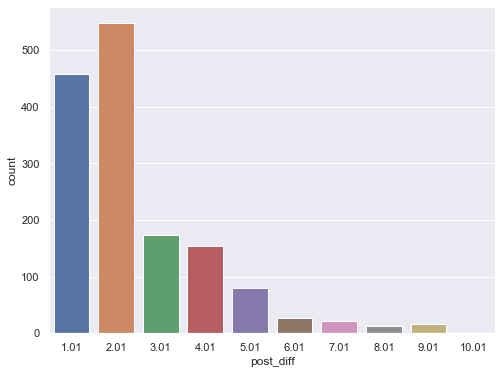


图 11 升职经历数据升职跨度统计柱状图

升职年份为升职后年份减升职前年份（next\_post\_year-current\_post\_year）。对升职经历数据中的升职年份进行统计，结果如下图所示。可以看出，升职年份在10年之内的数据占大多数。

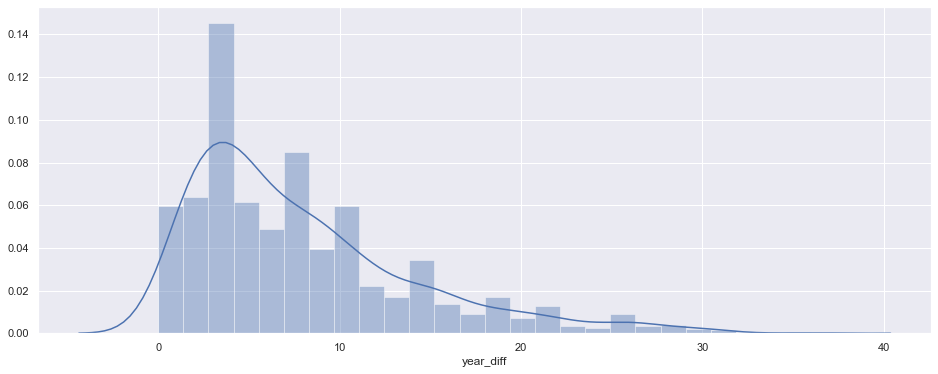


图 12 升职经历数据升职年份统计图

## 相关性分析

首先分析升职跨度与升职年份之间的关系，结果如下图所示。可以看出，数据中的数据大多是升职跨度较小且升职年份较短的，升职跨度和升职年份之间没有明显的相关性。

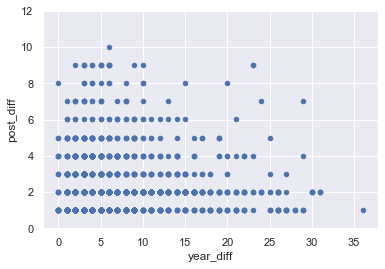


图 13 升职跨度与升职年份关系图

年龄取升职前的年龄（current\_post\_year-born\_year），分析年龄与升职年份之间的关系，结果如下图所示。可以看出，年龄和升职年份成负相关，年龄越大，升职年份越小。

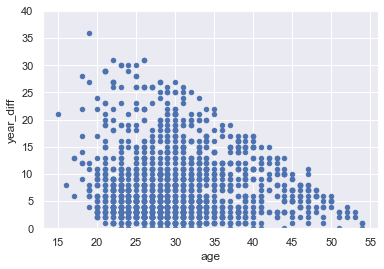


图 14 年龄与升职年份关系图

升职经历数据的热力图如下图所示。可以看出，出生年份、工作年份和入党年份这几个特征之间的关系较为密切。当前职位和升职职位之间关系密切。当前职位入职年份和升职职位入职年份之间关系密切。更值得注意的是，升值职位入职年份和年龄、累计工作时间、累计入党时间之间关系密切。

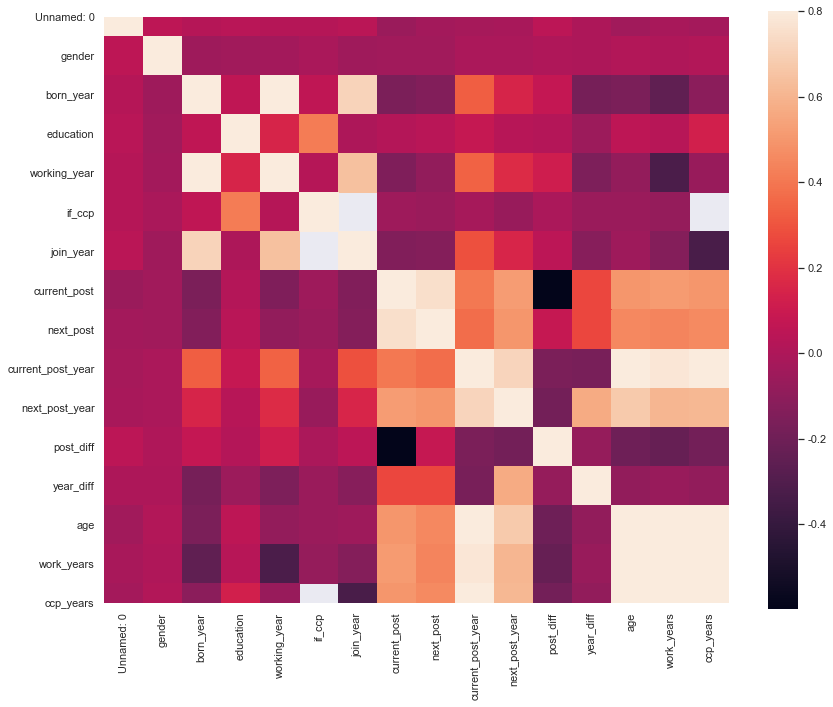


图 15 升职经历数据热力图

升职经历数据各特征之间相关性分析如下图所示。可以看出，年龄、累计工作时间、累计入党时间之间存在明显的正相关性。而升职所用时间，与累计工作时间、累计入党时间、年龄之间存在明显的负相关性。

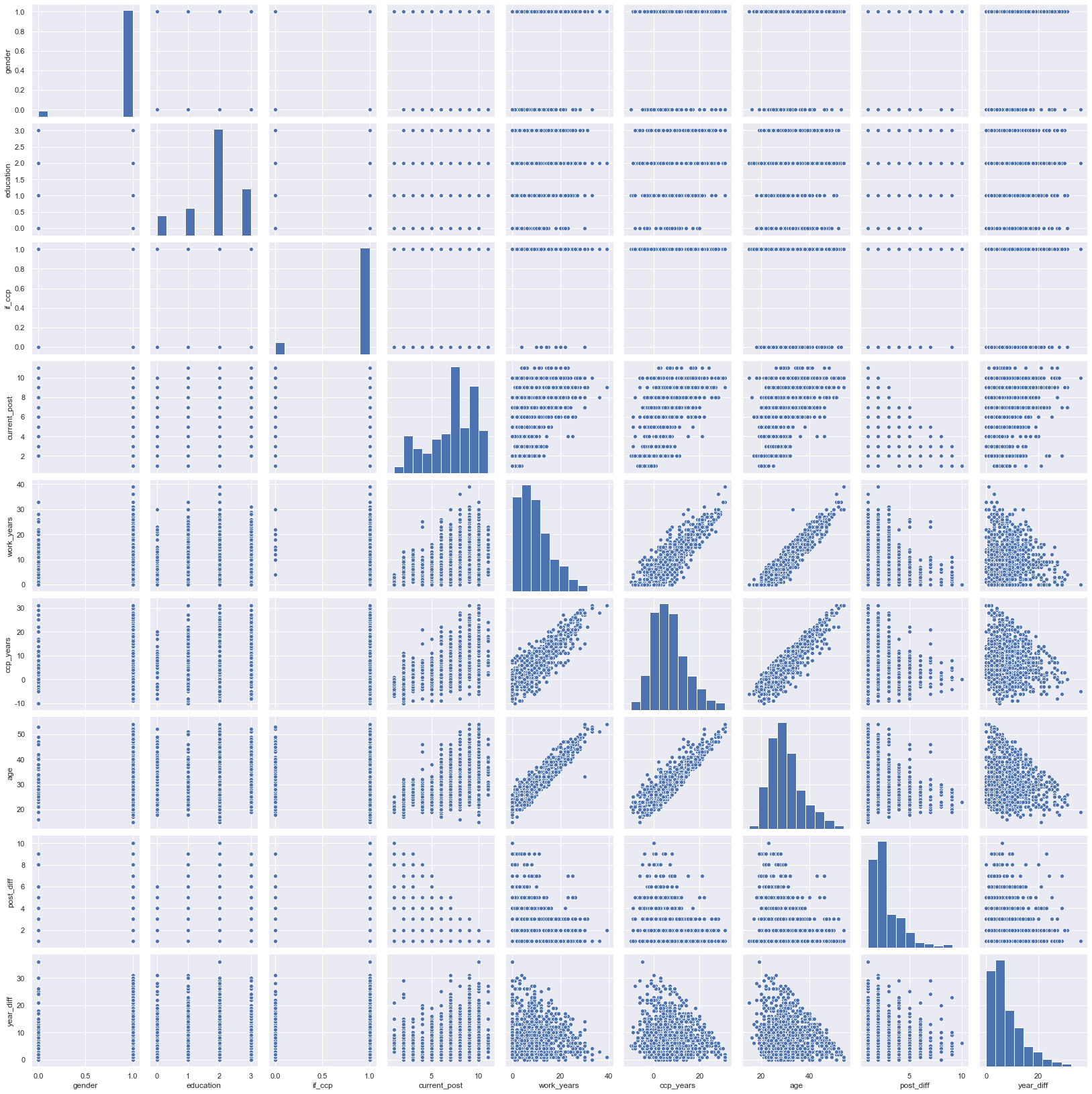


图 16 相关性分析图

## 缺失值处理

将升职经历数据中存在任一值缺失的数据直接删除。共有237条数据中的入党年份缺失，177条数据中的累计工作时间缺失，14条数据中的年龄缺失。

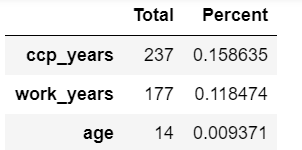


图 17 缺失数据统计

## 训练集和测试集生成

首先用sklearn包来对数据进行标准化处理。然后将20%的数据划分为测试集，80%的数据划分为训练集。

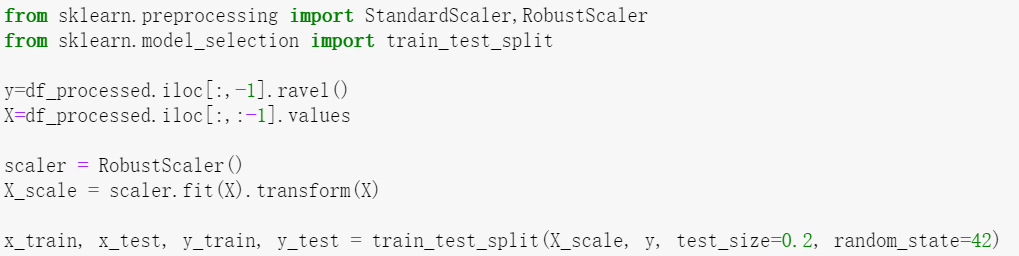


图 18 标准化和数据集划分代码

## 模型简介

本文使用了10个机器学习模型进行学习，并使用五折交叉验证进行评估。

* **Ridge** 和 **Lasso**

lasso 回归和RIdge回归（ridge regression）其实就是在标准线性回归的基础上分别加入 L1 和 L2 正则化（regularization）。

* **Random Forrest**

随机森林实际上是一种特殊的bagging方法，它将决策树用作bagging中的模型。是一个包含多个决策树的[分类器](https://baike.baidu.com/item/%E5%88%86%E7%B1%BB%E5%99%A8)， 并且其输出的类别是由个别树输出的类别的众数而定。

* **Gradient Boosting Tree**

梯度提升算法是一种通用的学习算法，除了决策树，还可以使用其它模型作为基学习器。梯度提升算法的思想是通过调整模型，让损失函数的值不断减小， 然后将各个模型加起来作为最终的预测模型。而梯度提升决策树则是以决策树为基学习器。

* **Support Vector Regression**

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。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。传统回归方法当且仅当回归f(x)完全等于y时才认为预测正确，如线性回归中常用(*f*(*x*)−*y*)2来计算其损失。而支持向量回归则认为只要f(x)与y偏离程度不要太大，既可以认为预测正确，不用计算损失

* **Stochastic Gradient Descent**

随机梯度下降在计算下降最快的方向时时随机选一个数据进行计算，而不是扫描全部训练数据集，这样就加快了迭代速度。随机梯度下降并不是沿着J(θ)下降最快的方向收敛，而是震荡的方式趋向极小点。

* **BayesianRidge**

贝叶斯岭回归，其跟[岭](https://link.jianshu.com/?t=http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html)相似。参数ω，α和λ都是在拟合模型的过程中同步估计出的。而剩余的超参数则是α和λ \*\*的先验参数。他们通常都是被选择为无信息的，

* **XgBoost**

XGBoost是一个树集成模型，他将K（树的个数）个树的结果进行求和，作为最终的预测值。

* **ElasticNet**

ElasticNet回归，综合了L1正则化项和L2正则化项，是一种使用L1和L2先验作为正则化矩阵的线性回归模型.这种组合用于只有很少的权重非零的稀疏模型

* **Extra-Trees**

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。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。

## 模型评估

我们使用交叉验证的均方根误差来评估模型的准确性。均方根误差是用来衡量真实值y和预测值之间的偏差，式子如下图所示。

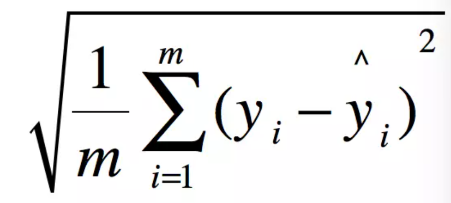


图 19 均方根误差

代码如下图所示。

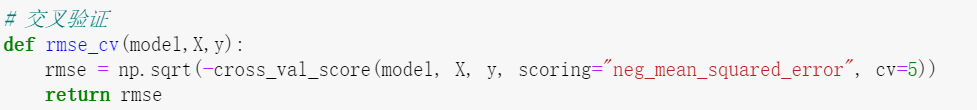


图 20 模型评估代码

## 模型训练

我们使用了sklearn包中的Ridge()、Lasso()、RandomForestRegressor()、GradientBoostingRegressor()、SVR()、LinearSVR()、ElasticNet()、SGDRegressor()、BayesianRidge()、KernelRidge()、ExtraTreesRegressor()、XGBRegressor()来进行训练。代码如下图所示。



图 21 模型训练代码

各个模型均采用默认参数，进行交叉验证，得到的均方根误差的平均值和标准差如下图所示。各模型的回归都能达到较好的效果，其中SGDRegressor()和BayesianRidge()模型回归得到的均方根误差的平均值最低。

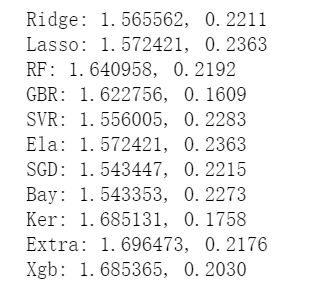


图 22 各模型默认参数的评估结果

### Ridge

下面我们对各个模型进行调参，首先是Ridge()模型。调参结果如下图所示。可以看出，当alpha取值为35时模型均方根误差的平均值最低。参数为35的Ridge模型在测试集上预测结果的均方根误差的平均值为1.5684。

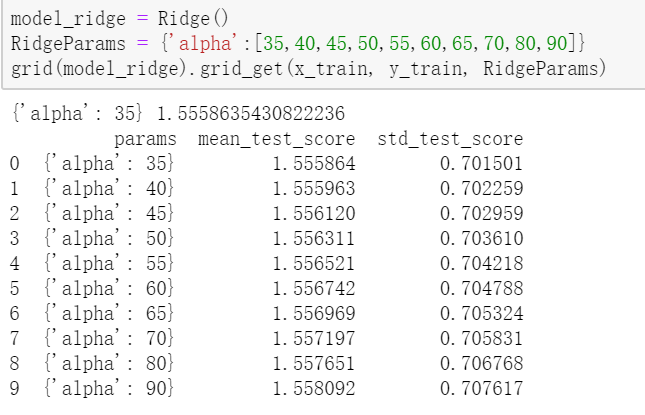


图 23 Ridge调参结果图

### Lasso

我们对Lasso()模型进行调参。调参结果如下图所示。可以看出，当alpha取值为0.01时模型均方根误差的平均值最低。参数为0.01的Lasso模型在测试集上预测结果的均方根误差的平均值为1.5779。

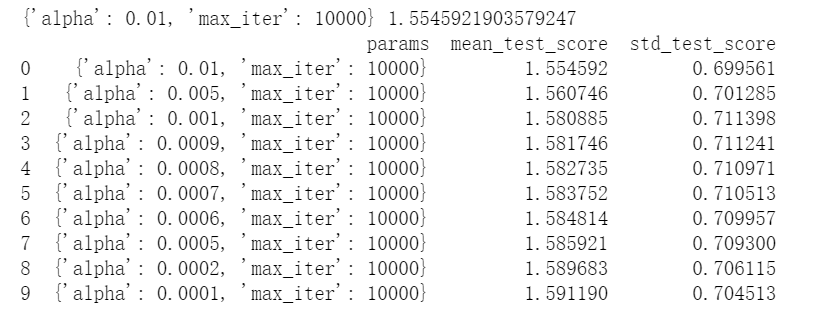


图 24 Lasso调参结果图

### RandomForestRegressor

调整RandomForestRegressor模型的n\_estimators参数。调参结果如下图所示。可以看出，当n\_estimators取值为780时模型均方根误差的平均值和标准差最低。

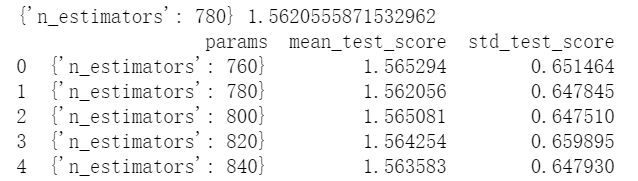


图 25 RandomForestRegressor调参结果图（1）

取n\_estimators的值为780，再对max\_depth和min\_samples\_split进行调整。max\_depth取6到13和min\_samples\_split取2到15的均方根误差如下图所示。可以看出，max\_depth取13和min\_samples\_split取9时均方根误差的平均值最低。N\_estimators为780、max\_depth为13和min\_samples\_split为9的RandomForestRegressor模型在测试集上预测结果的均方根误差的平均值为1.5187。

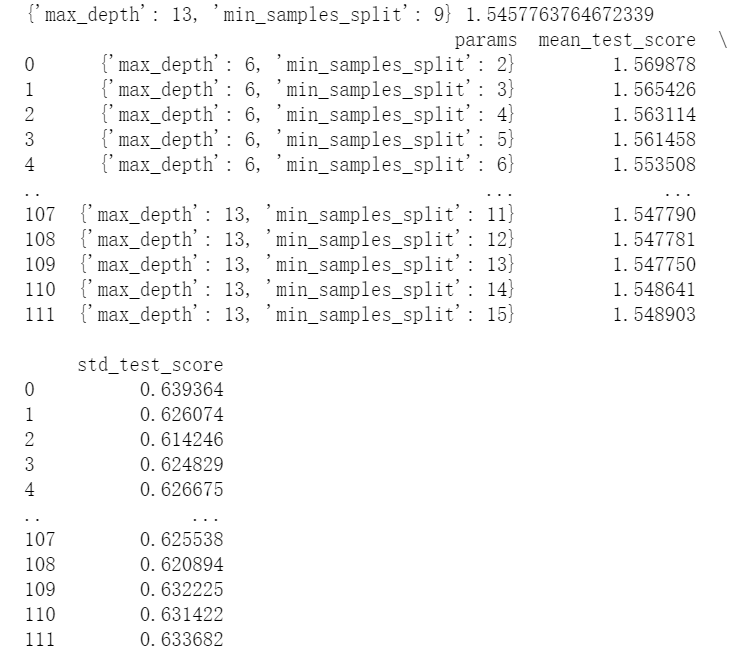


图 26 RandomForestRegressor调参结果图（2）

在RandomForestRegressor模型中对各个特征的重要性进行排序，结果如下图所示。可以看出，与升职年份最相关的特征是年龄，其次是入党年份和工作年份，出生地也对升职年份有一定的影响，而民族对升职年份几乎没有影响。

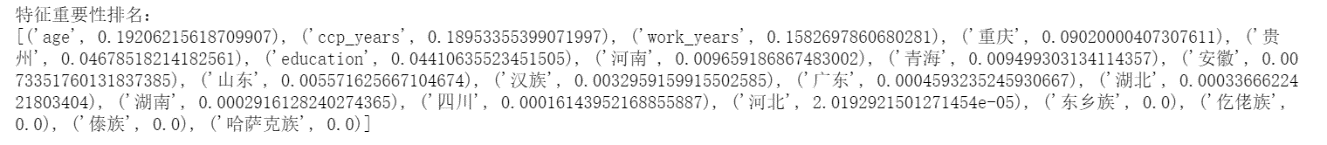


图 27 RandomForestRegressor中特征重要性排序结果

### BayesianRidge

调整BayesianRidge模型的alpha\_1、lambda\_1、alpha\_2、lambda\_2参数。先取alpha\_2为1e-6、lambda\_2为1e-6，调整alpha\_1、lambda\_1的取值，结果如下图所示。可以看出，模型的均方根误差几乎不受alpha\_1、lambda\_1的影响，因此我们取alpha\_1为1e-5、lambda\_1为1e-5。

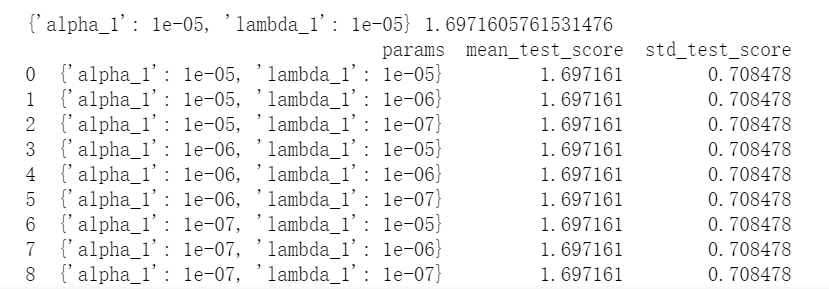


图 28 BayesianRidge调参结果图（1）

取alpha\_1为1e-5、lambda\_1为1e-5，再对alpha\_2和lamda\_2进行调整。alpha\_2和lamda\_2分别取1e-5、1e-6、1e-7的均方根误差如下图所示。可以看出，模型的均方根误差几乎不受alpha\_2、lambda\_2的影响，因此我们取alpha\_2为1e-5、lambda\_2为1e-5。alpha\_1、lambda\_1、alpha\_2、lambda\_2都为1e-5的BayesianRidge模型在测试集上预测结果的均方根误差的平均值为1.5655。

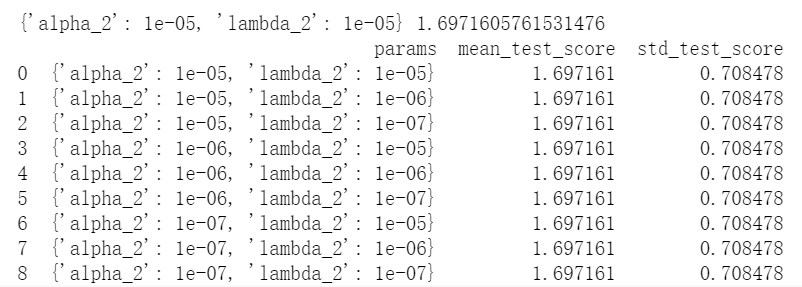


图 29 BayesianRidge调参结果图（2）

### SVR

调整SVR模型的degree、gamma、kernel参数。调参结果如下图所示。可以看出，当degree为2、gamma为scale、kernel为rbf时模型均方根误差的平均值最低。

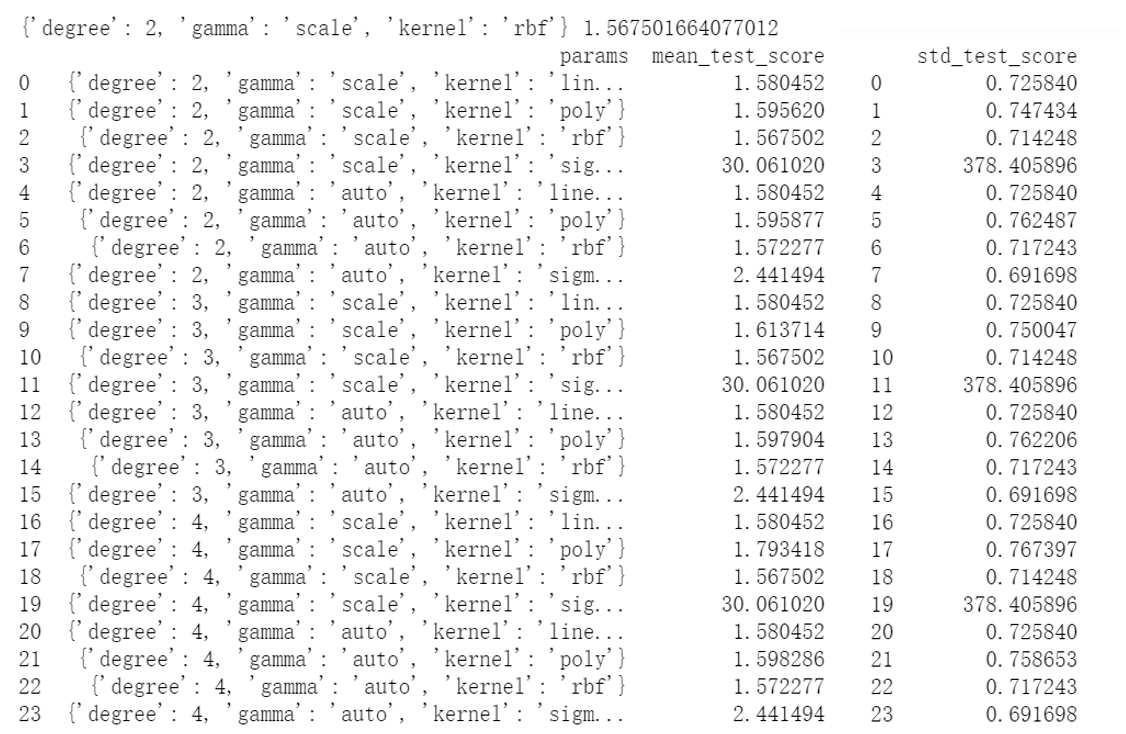


图 30 SVR调参结果图（1）

取degree为2、gamma为scale、kernel为rbf，再对C进行调整。不同C取值的SVR模型的均方根误差如下图所示。可以看出，模型的均方根误差几乎不受C取值的影响，因此我们取C为0.5。

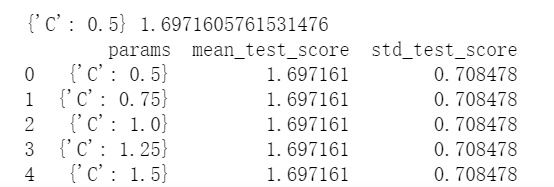


图 31 SVR调参结果图（2）

取degree为2、gamma为scale、kernel为rbf、C为0.5，SVR模型在测试集上预测结果的均方根误差的平均值为1.5722。

### GradientBoostingRegressor

调整GradientBoostingRegressor模型的loss参数。调参结果如下图所示。可以看出，当loss为huber时模型均方根误差的平均值最低，loss为lad时模型均方根误差的平均值同样也很低。

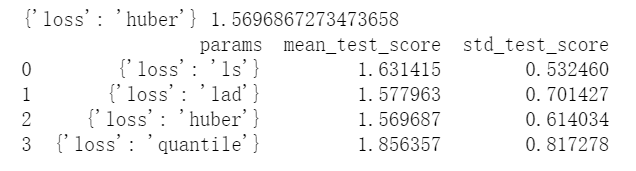


图 32 GradientBoostingRegressor调参结果图（1）

取loss为hunber，再对n\_estimators进行调整，结果如下图所示。可以看出，当n\_estimators的取值为90时模型均方根误差的平均值最低。

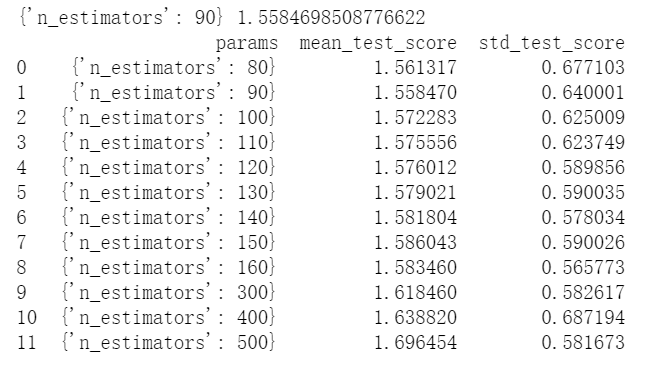


图 33 GradientBoostingRegressor调参结果图（2）

对max\_depth进行调整时发现loss取lad时效果更好，因此我们取loss为lad、n\_estimators为90，再对max\_depth进行调整，结果如下图所示。可以看出，当max\_depth的取值为90时模型均方根误差的平均值最低。

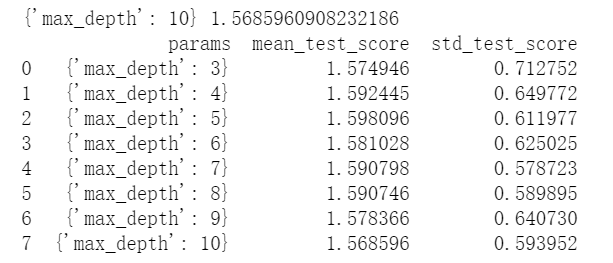


图 34 GradientBoostingRegressor调参结果图（3）

然后对min\_samples\_leaf、min\_samples\_split、min\_weight\_fraction\_leaf进行调整，结果如下图所示。

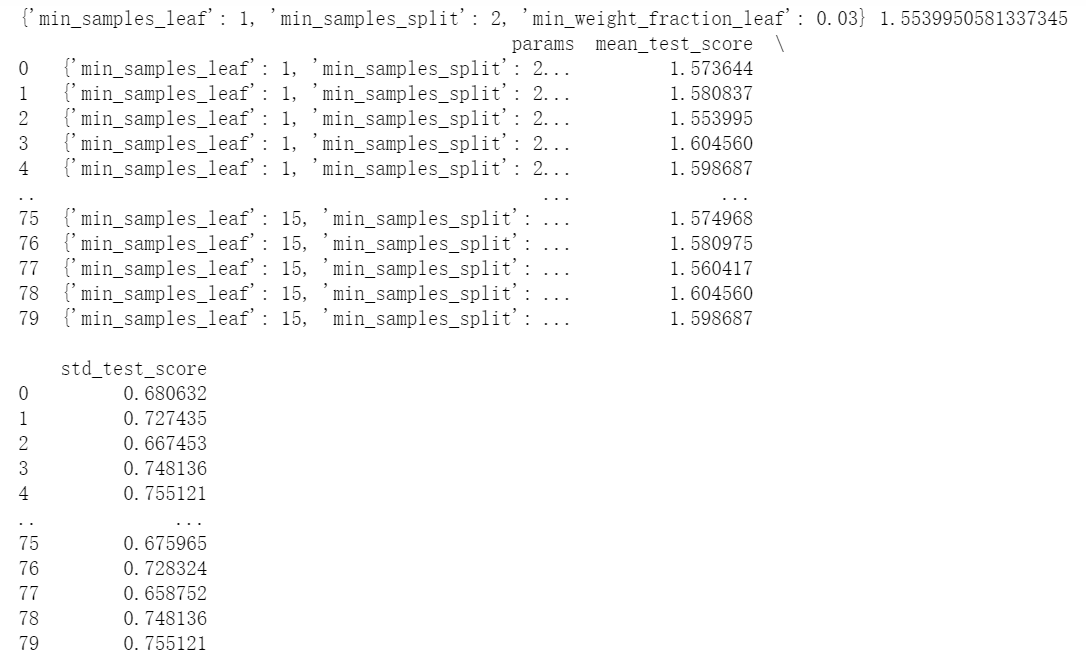


图 35 GradientBoostingRegressor调参结果图（4）

最后模型取loss为lad、n\_estimators为140、learning\_rate为0.1、max\_depth为5、subsample为1、alpha为0.9、min\_samples\_leaf为1、min\_samples\_split为2、min\_weight\_fraction\_leaf为0.03。GradientBoostingRegressor模型在测试集上预测结果的均方根误差的平均值为1.5376。

### ElasticNet

调整ElasticNet模型的alpha参数。调参结果如下图所示。可以看出，alpha对模型预测结果的均方根误差影响不大，当alpha为0.0018时模型的均方根误差的平均值最低。

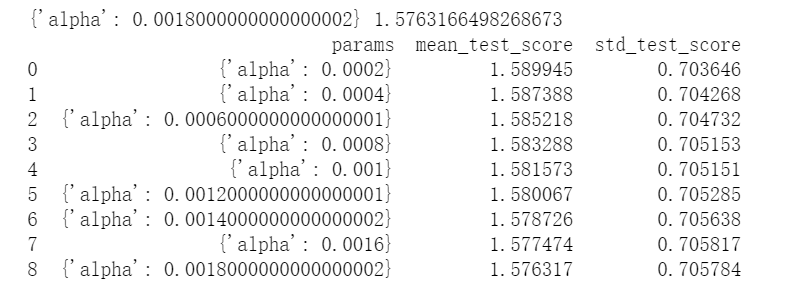


图 36 ElasticNet调参结果图（1）

取alpha为0.0012，调整模型的l1\_ratio参数，结果如下图所示。可以看出，当l1\_ratio取0.3时模型的效果最好。

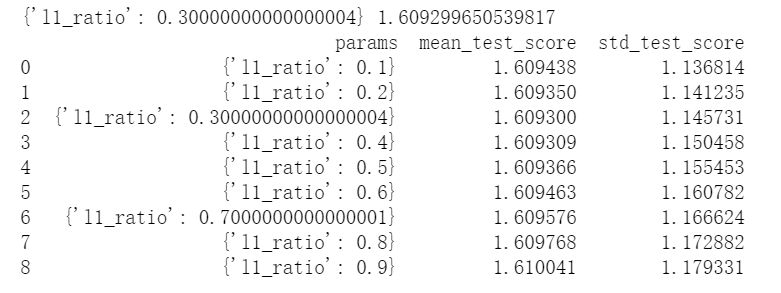


图 37 ElasticNet调参结果图（2）

当alpha为0.0012、l1\_ratio为0.3时模型在测试集上预测结果的均方根误差的平均值为1.6878。

### SGDRegressor

调整SGDRegressor模型的alpha参数，结果如下图所示。可以看出，alpha的取值对模型的均方根误差的平均值影响不大。

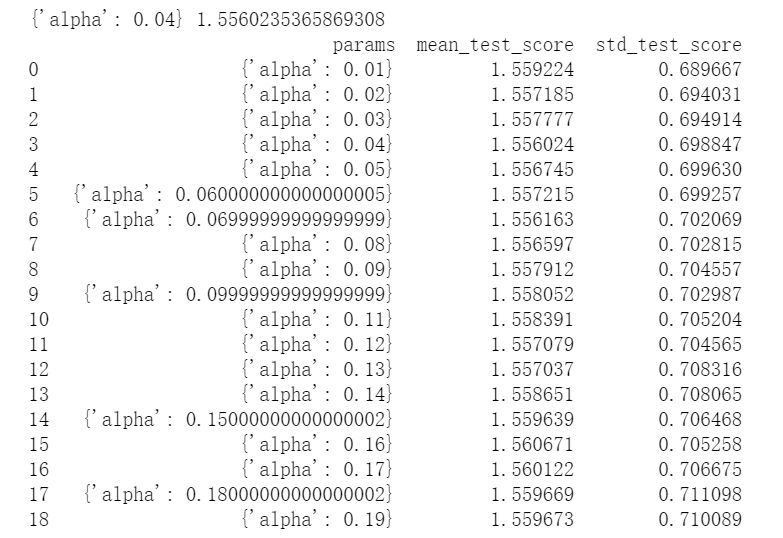


图 38 SGDRegressor调参结果图（1）

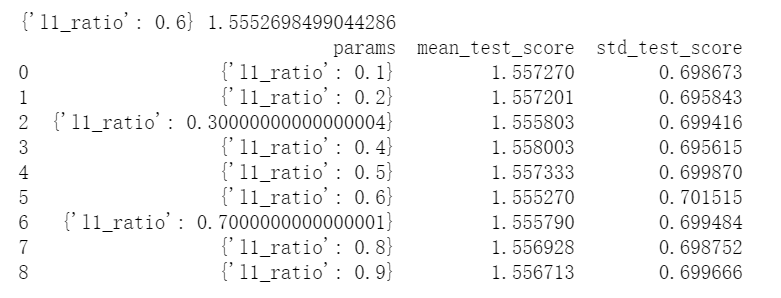
取alpha为0.04，调整l1\_ratio的取值，结果如下图所示。可以看出当l1\_ratio的取值为0.6时模型均方根误差的平均值最低。

图 39 SGDRegressor调参结果图（2）

取alpha为0.04、l1\_ratio为0.6，调整tol的取值，结果如下图所示。可以看出当tol的取值为1e-5时模型均方根误差的平均值最低。

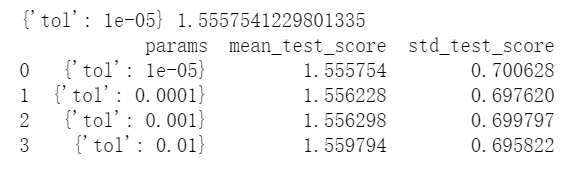


图 40 SGDRegressor调参结果图（3）

取alpha为0.04、l1\_ratio为0.6、tol为1e-5，调整max\_iter的取值，结果如下图所示。可以看出当max\_iter的取值为90000时模型均方根误差的平均值最低。

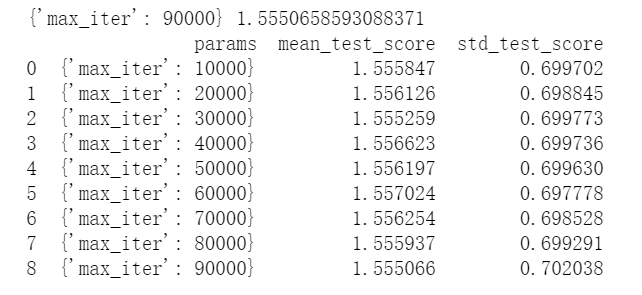


图 41 SGDRegressor调参结果图（4）

最后模型取alpha为0.04、l1\_ratio为0.6、tol为1e-5、max\_iter为90000。SGDRegressor模型在测试集上预测结果的均方根误差的平均值为1.5685。

### XGBRegressor

调整XGBRegressor模型的n\_estimators参数，结果如下图所示。可以看出，当n\_estimators为55时模型的均方根误差的平均值最小。

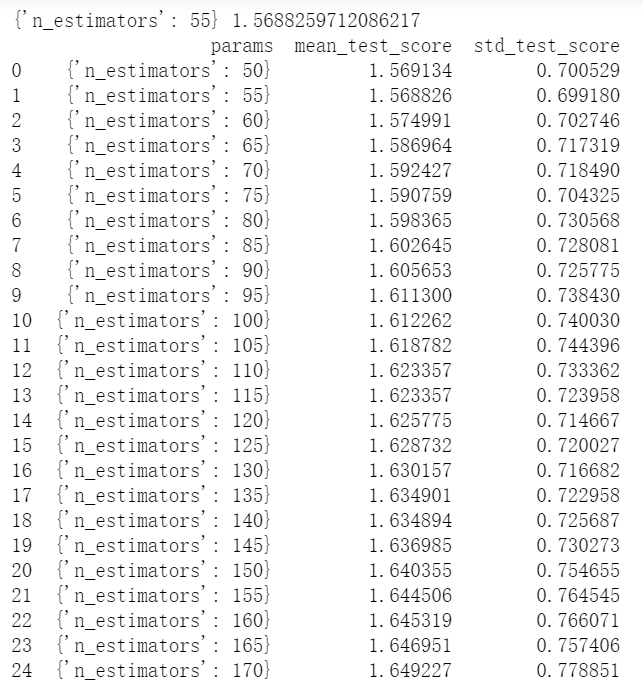


图 42 XGBRegressor调参结果图（1）

取n\_estimators为55，调整max\_depth和min\_child\_weight的取值，部分结果如下图所示。当max\_depth为10、min\_child\_weight为10时模型均方根误差的平均值最低。

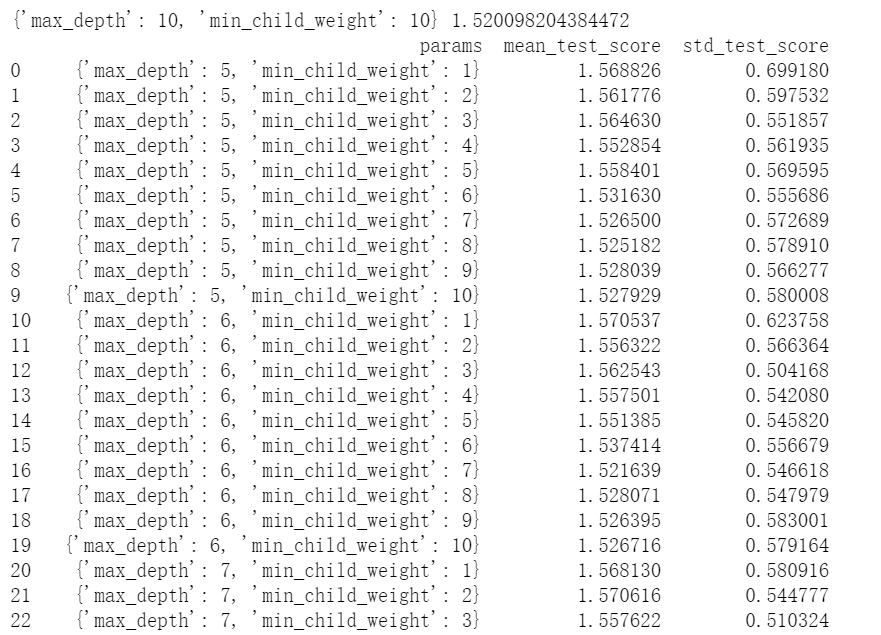


图 43 XGBRegressor调参部分结果图（2）

取n\_estimators为55、max\_depth为10、min\_child\_weight为10，调整gamma的取值，结果如下图所示。当gamma取0.1时模型均方根误差的平均值最低。

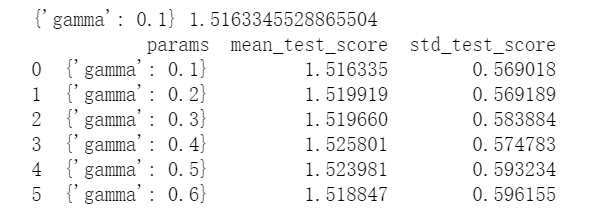


图 44 XGBRegressor调参结果图（3）

取n\_estimators为55、max\_depth为10、min\_child\_weight为10、gamma为0.1，调整colsample\_bytree和subsample的取值，结果如下图所示。当colsample\_bytree为0.8、subsample为0.8时模型均方根误差的平均值最低。

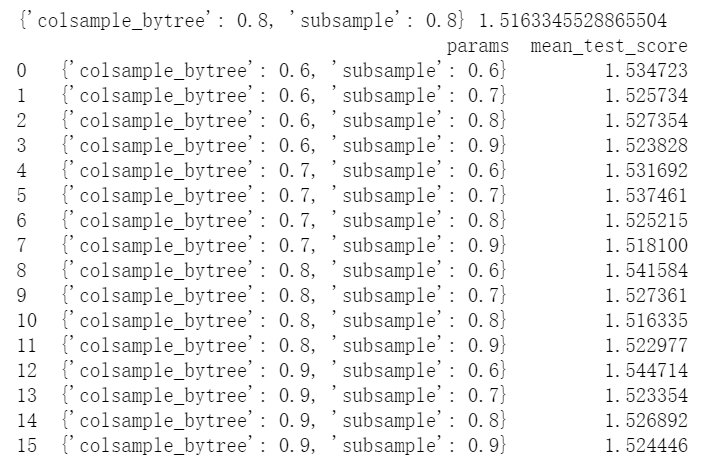


图 45 XGBRegressor调参结果图（4）

取n\_estimators为55、max\_depth为10、min\_child\_weight为10、gamma为0.1、colsample\_bytree为0.8、subsample为0.8，调整reg\_alpha和reg\_lambda的取值，结果如下图所示。当reg\_alpha为0.05、reg\_lambda为0.1时模型均方根误差的平均值最低。

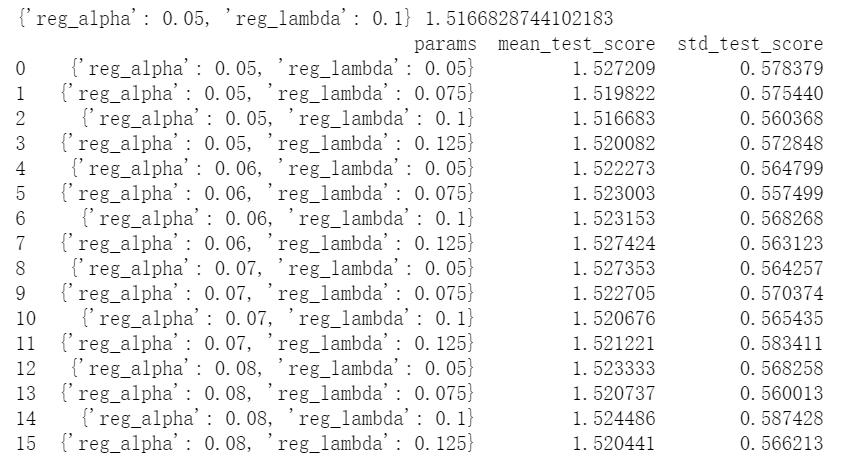


图 46 XGBRegressor调参结果图（5）

取n\_estimators为55、max\_depth为10、min\_child\_weight为10、gamma为0.1、colsample\_bytree为0.8、subsample为0.8、reg\_alpha为0.05、reg\_lambda为0.1，调整learning\_rate的取值，结果如下图所示。当learning\_rate为0.08时模型均方根误差的平均值最低。

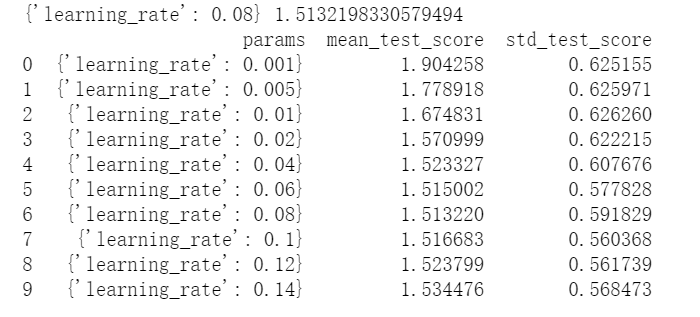


图 47 XGBRegressor调参结果图（6）

最后模型取n\_estimators为55、max\_depth为10、min\_child\_weight为10、gamma为0.1、colsample\_bytree为0.8、subsample为0.8、reg\_alpha为0.05、reg\_lambda为0.1、learning\_rate为0.08。XGBRegressor模型在测试集上预测结果的均方根误差的平均值为1.5854。

### ExtraTreesRegressor

调整ExtraTreesRegressor模型的max\_depth和min\_sample\_split参数，部分结果如下图所示。当max\_depth为10、min\_sample\_split为20时模型的均方根误差的平均值最小。

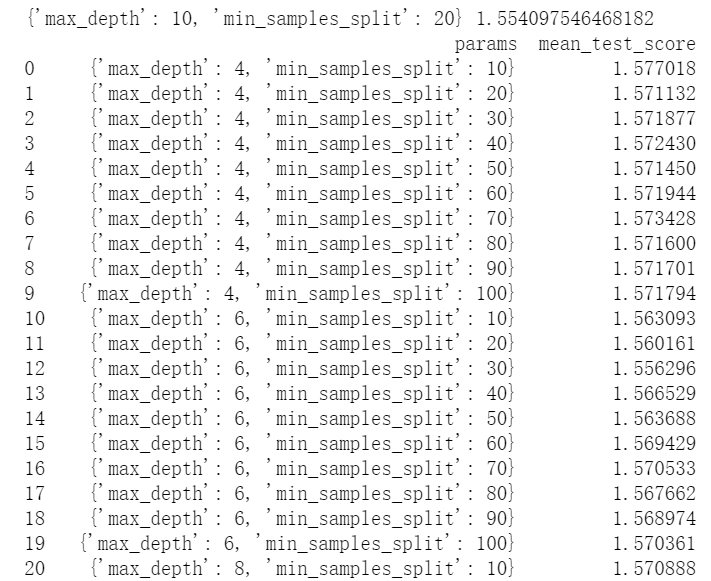


图 48 ExtraTreesRegressor调参部分结果图（1）

取max\_depth为10、min\_sample\_split为20，调整min\_samples\_leaf的取值，结果如下图所示。当min\_samples\_leaf为1时模型均方根误差的平均值最低。

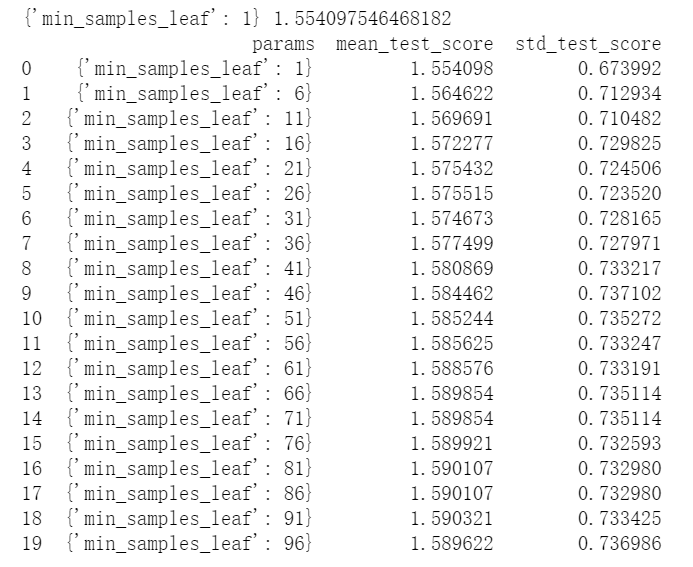


图 49 ExtraTreesRegressor调参结果图（2）

最后模型取max\_depth为10、min\_sample\_split、min\_samples\_leaf为1。ExtraTreesRegressor模型在测试集上预测结果的均方根误差的平均值为1.5854。

## 模型比较

下面我们比较上述10个模型在测试集上预测结果的均方根误差的平均值大小。结果如下图所示。可以看出，ElasticNet预测结果的均方根误差均值最大，模型效果最差；RandomForestRegressor预测结果的均方根误差均值最小，模型效果最好。

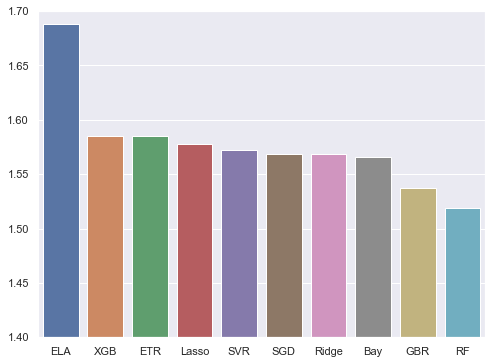


图 50 模型比较图

## 模型集成

选取上述模型中结果较好的5个模型进行集成。0.35、0.25、0.15、0.15、0.1分别为RandomForestRegressor、GradientBoostingRegressor、BayesianRidge、Ridge、SGDRegressor的比例权重。

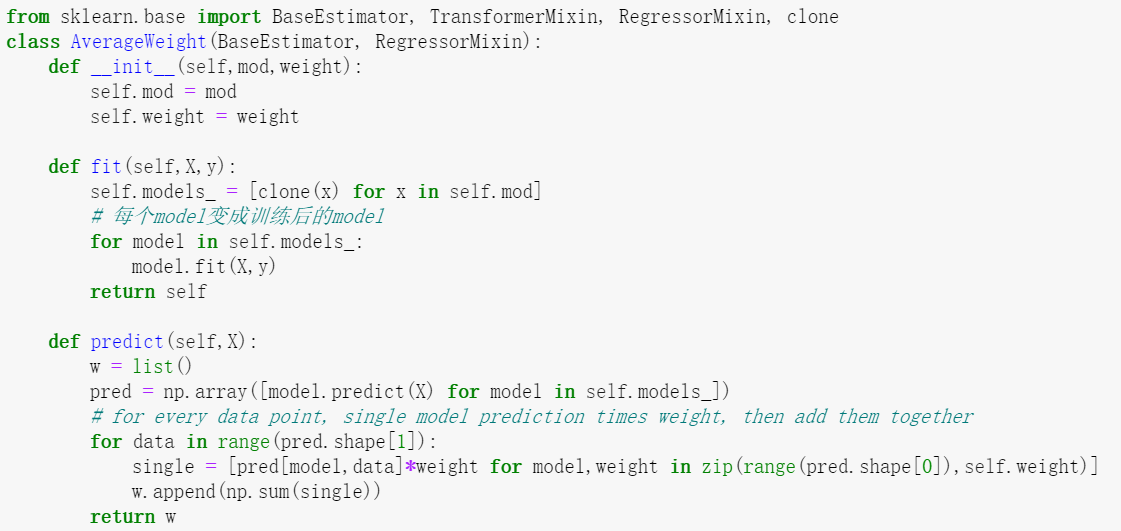


图 51 模型集成代码示例

集成模型在训练集上进行交叉验证的均方根误差的平均值为1.5306，低于单个模型，但在测试集上预测结果的均方根误差为1.6076，高于单个模型。

## Stack模型集成

使用Stack的方法对表现最好的五个模型进行集成。代码示例如下图所示。

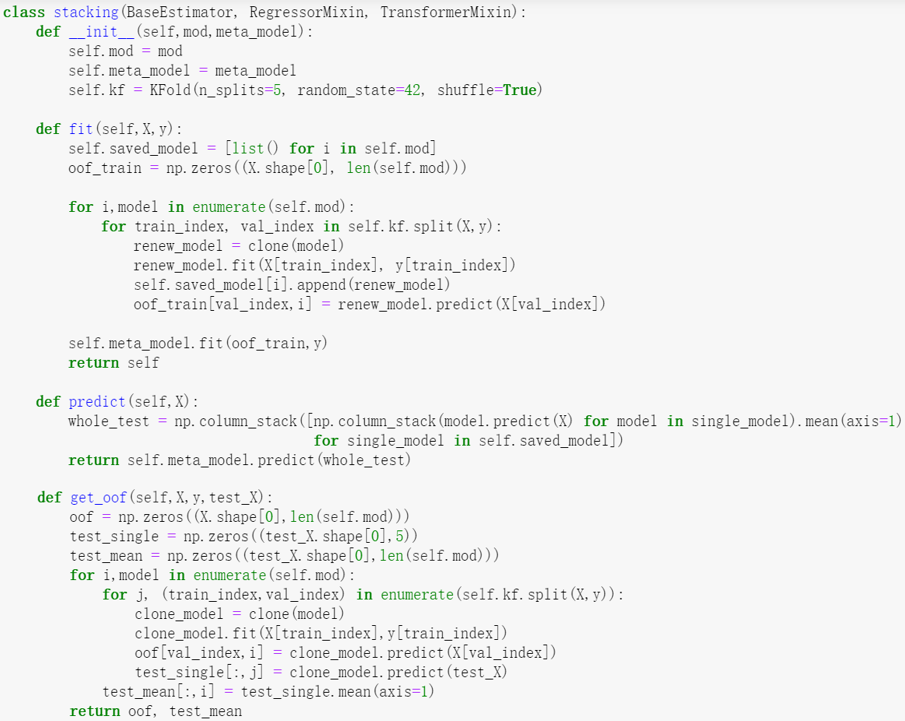


图 52 Stack集成代码示例

Stack集成模型在训练集上进行交叉验证的均方根误差的平均值为1.5339，略高于单个模型，在测试集上预测结果的均方根误差为1.5325，并不明显低于单个模型。

# 项目总结

本文在数据集上使用了10个模型进行训练和预测，在数据集上产生了良好的实验结果，接着使用2种集成方法集成了上述10个模型中结果较好的5个模型，但上述两种方法并不能对上述模型效果进行提升。综上所述，选取上述模型中实验结果最好的RandomForestRegressor作为官员晋升时长的预测，通过该模型我们还能获得各个特征的重要性。

在平均个人每个阶段晋升需要8年时间的情况下，通过本文的特征提取和模型预测，能在误差大概1.5年左右，通过一个人的基本信息和所处的岗位，预测到下一次晋升需要的时间，发掘出了每个官员的潜力。

# 小组分工

**数据爬取**： 周彧祺

**数据预处理**： 何金鸿、周彧祺

**特征相关性分析**： 周彧祺、刘珏

**机器学习模型训练**： 周彧祺、何金鸿、吕美林、刘珏、王木

**多模型集成**： 周彧祺、吕美林

**文档书写**： 刘珏、周彧祺