# 校名

**模式识别**

作 业 kaggle房价预测

年 级 2019级

学 号 Y30190729

姓 名 李江坤

指导老师 赵海涛

2019年 12 月 1日

# 基于集成学习的房价预测建模

# 摘要

房价预测成为机器学习与深度学习一个常见的应用场景。本文对kaggle比赛房价预测进行基于统计学知识进行建模，分析影响影响房价的各个特征，建立特征之间的相关系数，对原始数据集进行均值填补，剔除丢失过多的特征，筛选出与房价相关系数最佳的特征，将数据进行独热编码，采用进行数据处理过后的数据，结合ElasticNet回归与RandomForestRegressor算法进行建模，最后采用集成学习预测房价输出。

# 数据集

Kaggle competition 的SalePrice 项目，其中训练数据和测试数据分别各有1460条，1459条。数据的特征列有80个，其中38个是数值类型的，42个类别类型。特征的详细描述在赛事链接里有介绍。

<https://www.kaggle.com/c/house-prices-advanced-regression-techniques>

# 预测模型的选择

# ElasticNet回归：

ElasticNet回归属于Lasso回归和岭回归的组合。

准则函数：其中（是正则参数，是回归参数）。

模型调用：调取scikit learn 中的sklearn.linear\_model.ElasticNetCV模型，模型具体参数: (l1\_ratio=0.5, eps=0.001, n\_alphas=100, alphas=None, fit\_intercept=True,normalize=False,precompute=’auto’,max\_iter=1000, tol=0.0001, cv=’warn’, copy\_X=True, verbose=0, *n\_jobs=None*, positive=False, random\_state=None, selection=’cyclic’)

其中本实验使用到模型参数的解释：

1. l1\_ratio: 0<l1\_ratio<1,表示正则化参数，可以设置为单值,例如l1\_ratio=0.1，也可以设置为一个数组l1\_ratio=[0.1 0.2 0.5 0.9],当设置为一个数组时，目的是在模型训练时可以通过交叉验证得到较好的正则参数，本实验采用l1\_ratio=[.01, .1, .5, .9, .99]。
2. alphas：alphas为一个数组，用来计算模型，本实验设计alphas =[0.0001, 0.0005, 0.001, 0.01, 0.1, 1, 10]。
3. max\_iter：模型训练时的迭代次数，本实验设置 max\_iter=5000。

# RandomForestRegressor随机森林回归：

随机森林由许多决策树构成，每个决策树生成的结果构成最后随机森林输出预测结果的一部分，最终结果输出依靠于每个决策树的投票选择，类似一种投票机制。

决策树（Decision Tree）：

决策树（Decision Tree）是一种非参数的有监督学习方法，它能够从一系列有特征和标签的数据中总结出决策规则，并用树状图的结构来呈现这些规则，以解决分类和回归问题。

决策树包含内部结点和叶结点，每个内部结点代表对每个不同特征（属性）的测试，每个叶结点输出代表一个种类。

决策树特征选择准则：信息增益。

信息增益：特征A对训练数据集D的信息增益g(D,A),定义为集合D的经验熵H(D)与特征A给定条件下D的经验条件熵H(D|A)之差，即：



熵（entropy）：



条件熵：表示在选定特征X条件下随机变量Y的不确定性



模型调用：调取scikit learn 中的sklearn.ensemble.RandomForestRegressor(n\_estimators=10, criterion='mse', max\_depth=None, min\_samples\_split=2, min\_samples\_leaf=1, min\_weight\_fraction\_leaf=0.0, max\_features='auto', max\_leaf\_nodes=None, bootstrap=True, oob\_score=False, n\_jobs=1, random\_state=None, verbose=0, warm\_start=False)。

其中本实验使用到模型参数的解释：

n\_estimators：设置随机森林决策树数目

max\_features：随机森林允许单个决策树使用特征的最大数量

# 实验过程：

# 数据处理需要的库

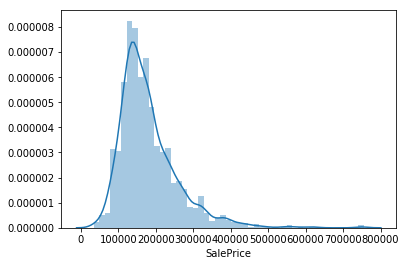
|  |
| --- |
| *#导入本次实验所需要用到的python库*  import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  import seaborn as sns  from sklearn import ensemble, tree, linear\_model  from sklearn.ensemble import RandomForestRegressor  from scipy import stats  from scipy.stats import norm  from sklearn.preprocessing import StandardScaler  from sklearn.model\_selection import train\_test\_split,cross\_val\_score |

# 读取数据集

|  |
| --- |
| train\_data=pd.read\_csv('/kaggle/input/train.csv')  test\_data=pd.read\_csv('/kaggle/input/test.csv') |

# 初步了解价格分布信息

|  |
| --- |
| train\_y = train\_data.pop('SalePrice')  y\_plot = sns.distplot(train\_y) |

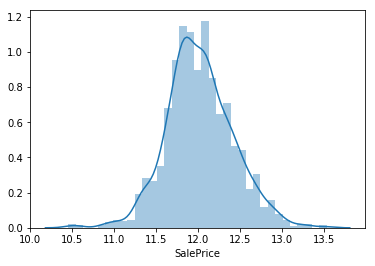


分析房价分布数据的Skewness和Kurtosis

|  |
| --- |
| Skewness=1.882876  Kurtosis=6.536282 |

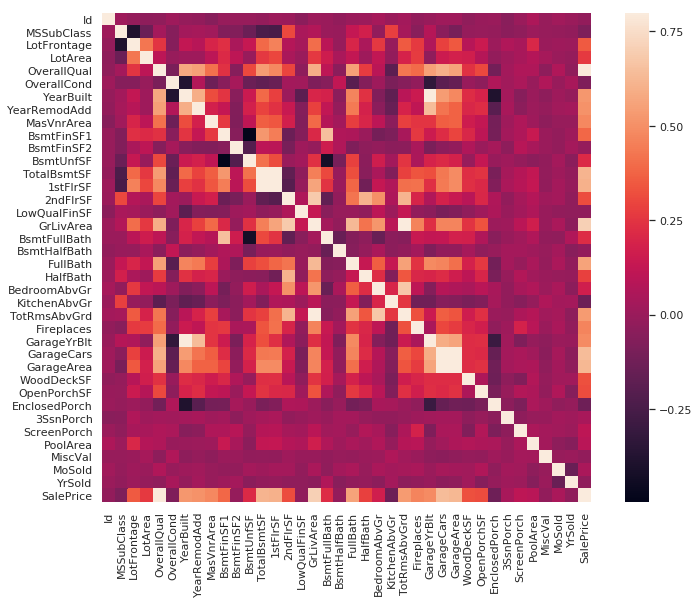
对房价取对数，调整房价数值大小，而不改变房价的分布状况

|  |
| --- |
| train\_y\_log = np.log(train\_y)  y\_plot\_log = sns.distplot(train\_y\_log) |



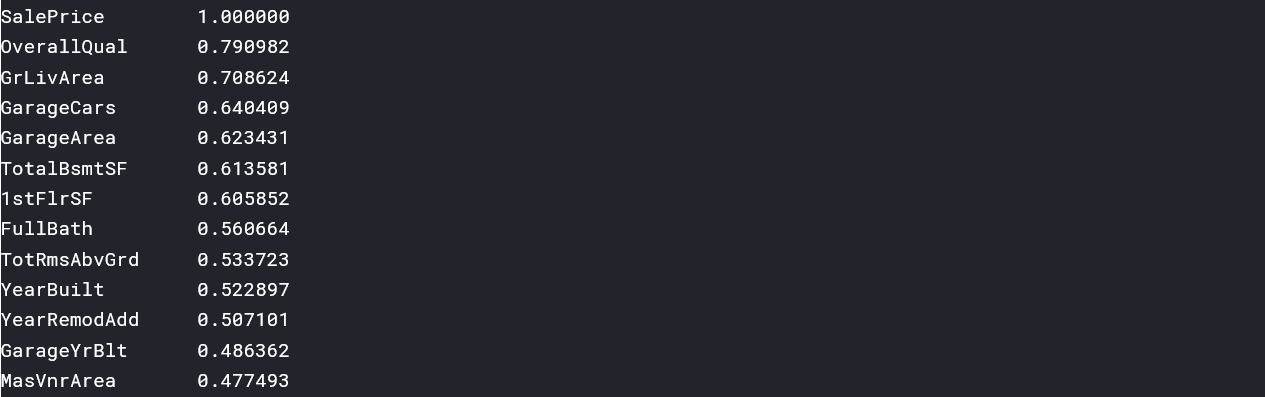
各个特征的相关性热力图

|  |
| --- |
| #correlation matrix  corrmat = train\_data.corr()  f, ax = plt.subplots(figsize=(12, 9))  sns.heatmap(corrmat, vmax=.8, square=True); |



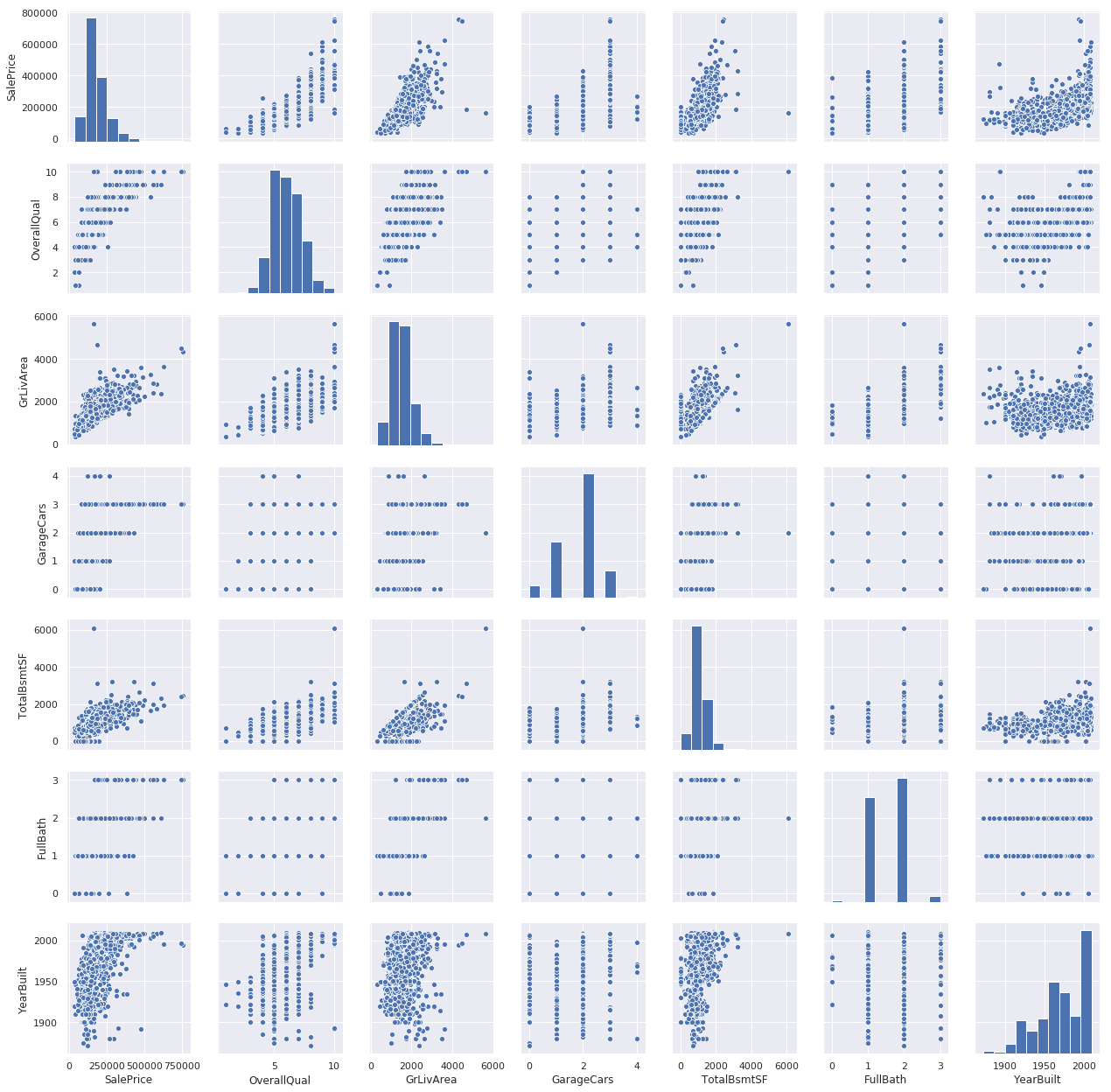
通过热力图筛选出与价格相关性较高的特征

|  |
| --- |
| corrmat\_rank=corrmat.sort\_values(by=['SalePrice'] , ascending=False)  corrmat\_rank.loc[:,'SalePrice'] |



筛选后特征分析图

|  |
| --- |
| sns.set()  cols=['SalePrice', 'OverallQual', 'GrLivArea', 'GarageCars', 'TotalBsmtSF', 'FullBath', 'YearBuilt']  sns.pairplot(train\_data[cols], size = 2.5)  plt.show(); |



将重要特征数据提取进行数据处理：

1. 填补丢失较少数据的特征信息
2. 剔除过滤丢失数据较多的特征
3. 对特征进行独热编码（one-hot-encode）

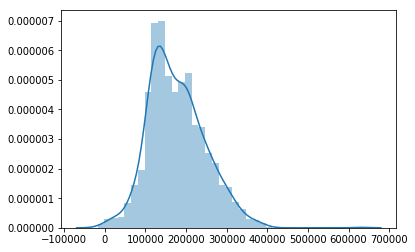
# 模型代码实现：

以上选取模型调用sk learn函数库中的‘linear\_model.ElasticNetCV（）’和‘RandomForestRegressor（）’两个训练模型。

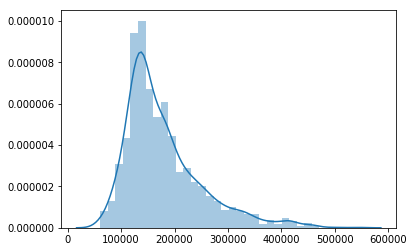
模型的训练参数alpha和Max\_feature都采用交叉验证选取，模型迭代次数5000次。

|  |
| --- |
| alphas = [0.0001, 0.0005, 0.001, 0.01, 0.1, 1, 10]  ElasticN=linear\_model.ElasticNetCV(alphas=([alpha], l1\_ratio=[.01, .1, .5, .9, .99], max\_iter=5000)  Model\_1 = ElasticN.fit(train\_x, train\_y) |
| Max\_features = [.1,.3,.5,.7,.9,.99]  RFR=RandomForestRegressor(n\_estimators=200,max\_features= feature)  Model\_2= RFR.fit(train\_x,train\_y) |

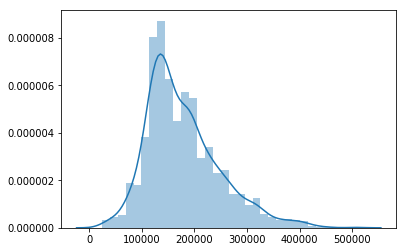
得到上述两个预测模型Model\_1和Model\_2,最后采用集成学习，将两个模型预测结果取平均，作为最后预测值。其预测结果如下：



Model\_1预测输出



Model\_2预测输出



集成预测输出

# 总结

通过本次kaggle比赛学习了Lasso回归和Ridge回归两种建模预测方法，其两者建模思想基于最小二乘法，通过引入不同范数的正则化惩罚项而减少过拟合现象，上述建模方法适用于特征数目多于已有数据的数目，特征维数较高，而数据数目相对不足。最后采用集成学习法将随机森林预测结果与ElasticNet回归预测结合，一定程度提高预测精度。最后由衷地感谢赵老师的模式识别课程，传授学生知识，引导学生探索和实践，学生铭记在心。

参考文献：

[1]王俊迪,许蕴山,彭芳,肖冰松.基于岭回归的红外协同定位优化算法[J/OL].北京航空航天大学学报:1-10[2019-12-06].

[2]赵川,揭海华,杨浩雄.基于遗传优化岭回归的GM(2,1)供应链风险预测[J].统计与决策,2018,34(10):51-53.

[3]薛仁政,陈淑鑫,黄宏本.A型恒星光谱线指数岭回归有效温度的预测分析[J].光谱学与光谱分析,2019,39(08):2624-2629.

[4]陈静,张艳新,姜媛媛.融合多特征与随机森林的纹理图像分类方法[J/OL].传感器与微系统,2019(12):58-61[2019-12-06].https://doi.org/10.13873/J.1000-9787(2019)12-0058-04.

[5]Salas Eric Ariel L,Subburayalu Sakthi Kumaran. Correction: Modified shape index for object-based random forest image classification of agricultural systems using airborne hyperspectral datasets.[J]. PloS one,2019,14(9).

重要变量：（对房价有重要影响的variable）

* OverallQual.(房屋质量)
* YearBuilt.（建造年份）
* TotalBsmtSF.（地下室总面积）
* GrLivArea.（地面以上居住面积平方英尺）

# Python绘图库对上述特征和目标变量的关系进行直观

|  |
| --- |
| train\_data=pd.read\_csv(‘F:\研究生1卷\Kaggle\SalePrice\SalePrice\Data(Original)\test\_datacsv')  figure=plt.figure()  sns.pairplot(x\_vars=['OverallQual','GrLivArea','YearBuilt','TotalBsmtSF'],y\_vars=['SalePrice'],  data=train\_data,dropna=True)  plt.show() |

# 独热编码one-hot code

独热码，在英文文献中称做 one-hot code, 直观来说就是有多少个状态就有多少比特，而且只有一个比特为1，其他全为0的一种码制。举例如下：

假如有三种颜色特征：红、黄、蓝。 在利用机器学习的算法时一般需要进行向量化或者数字化。那么你可能想令 红=1，黄=2，蓝=3. 那么这样其实实现了标签编码，即给不同类别以标签。然而这意味着机器可能会学习到“红<黄<蓝”，但这并不是我们的让机器学习的本意，只是想让机器区分它们，并无大小比较之意。所以这时标签编码是不够的，需要进一步转换。因为有三种颜色状态，所以就有3个比特。即红色：1 0 0 ，黄色: 0 1 0，蓝色：0 0 1 。如此一来每两个向量之间的距离都是根号2，在向量空间距离都相等，所以这样不会出现偏序性，基本不会影响基于向量空间度量算法的效果。

|  |
| --- |
| sklearn的例子  from sklearn import preprocessing  enc = preprocessing.OneHotEncoder()  enc.fit([[0, 0, 3], [1, 1, 0], [0, 2, 1], [1, 0, 2]])# fit来学习编码  enc.transform([[0, 1, 3]]).toarray() # 进行编码 |

优点：

独热编码（哑变量 dummy variable）是因为大部分算法是**基于向量空间**中的度量来进行计算的，为了使非偏序关系的变量取值不具有偏序性，并且到圆点是等距的。使用one-hot编码，将离散特征的取值扩展到了欧式空间，离散特征的某个取值就对应欧式空间的某个点。将离散型特征使用one-hot编码，会让特征之间的**距离计算更加合理**

缺点:

当类别的数量很多时，**特征空间**会变得非常大。在这种情况下，一般可以用PCA来**减少维度**。而且one hot encoding+PCA这种组合在实际中也非常有用。