**数据挖掘入门项目——房价预测**

* **一、数据可视化探索**

1. 探索数据质量（缺失值、异常值等）
2. 探索特征与预测变量之间的相关性

（1）先来导入数据，看一下数据的格式：

# 所需的工具包放在最前边

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# 1.先查看一下数据的格式

trainData = pd.read\_csv('train.csv')

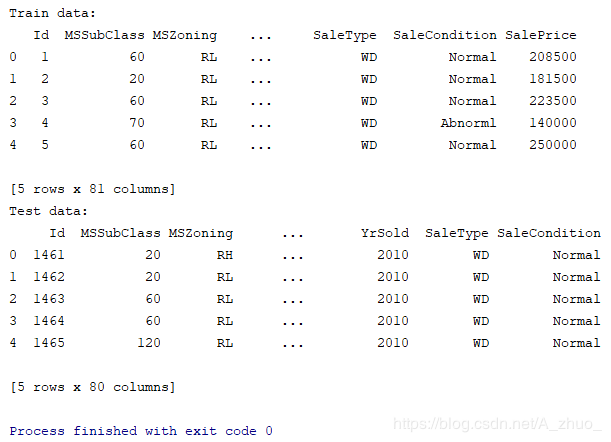
testData = pd.read\_csv('test.csv')

print("Train data:")

print(trainData.head(5))

print("Test data:")

print(testData.head(5))

结果：  
  
可以看到训练集：5行81列；测试集5行80列。训练集最后一列为房价。

（2）探索缺失值

# 2.查找缺失值

# 2.1 合并训练集和测试集一块处理数据，减少工作量

n\_trainData = trainData.shape[0] # 记录下训练集的行数

n\_testData = testData.shape[0] # 记录下测试集的行数

y\_train = trainData.SalePrice.values # 训练集最后一列（房价）

all\_data = pd.concat([trainData,testData])

# drop函数：剔除表中的列或者行。axis默认0表示删除行；inplace=true表示使用剔除掉列之后的数据替换原表，默认为false

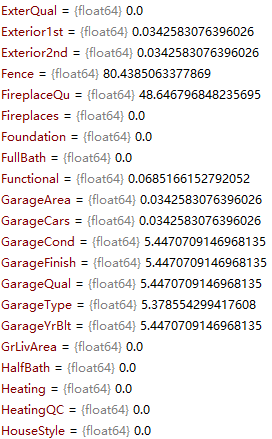
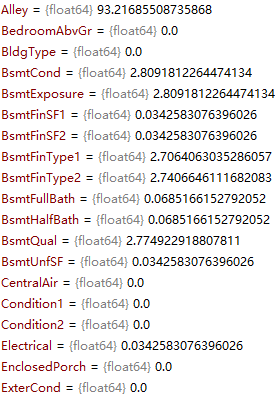
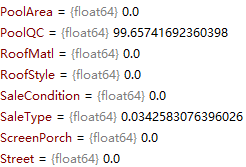
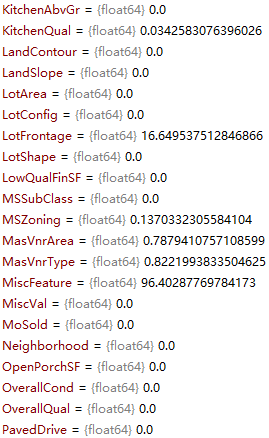
all\_data.drop(['SalePrice','Id'],axis=1,inplace=True) # 剔除掉Id列和房价列

print("all\_data's shape：{}".format(all\_data.shape))

在这里插入图片描述

# 2.2 计算缺失率

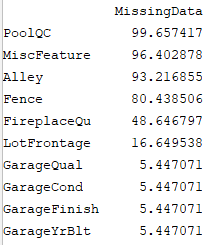
all\_data\_na = (all\_data.isnull().sum()/len(all\_data))\*100

all\_data\_na = all\_data\_na.drop(all\_data\_na[all\_data\_na==0].index).sort\_values(ascending=False) # 剔除掉缺失率为0的特征索引

missing\_data = pd.DataFrame({'MissingData':all\_data\_na})

print(missing\_data.head(10))



# 2.3 可视化缺失情况

x = all\_data\_na.index

y = all\_data\_na

plt.xticks(rotation=90)

plt.bar(x,y)

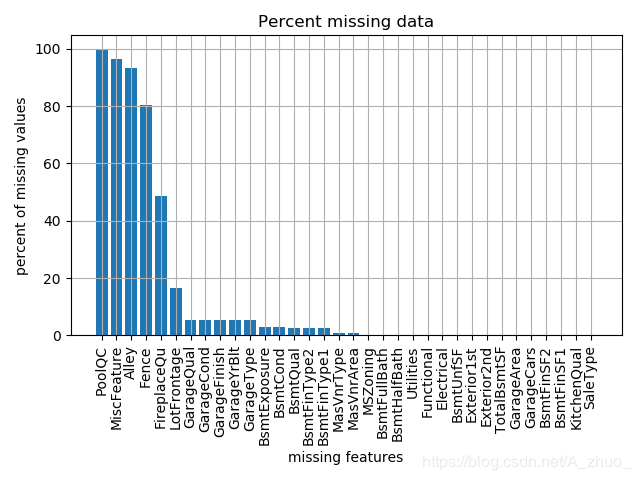
plt.xlabel('missing features')

plt.ylabel('percent of missing values')

plt.title('Percent missing data')

plt.grid()

plt.show()

  
可以看出PoolQC、MiscFeature和Alley三个特征值缺失率超过80%，后续考虑可能需要剔除掉这三个特征。其余缺失值的处理后续再做。

（3）探索异常值  
一般情况下，可以采用描述性统计查看是否存在异常值。第二，对于连续型变量可以采用散点图，查看特征与预测变量间的相关性的同时，发现异常值的存在；对于离散变量，可以采用画箱线图的方式，查看其特征分布，根据箱线图判断异常值。第三，可以采用3倍的标准差原则进行判断。  
这里，我们先查看特征与目标变量间的相关性，选择相关性大的特征探索异常值，这样才具有研究的价值吧（我是这样考虑的哈）。

# 3.查找异常值

# 3.1 探索变量间的相关性（以热力图的形式展现）

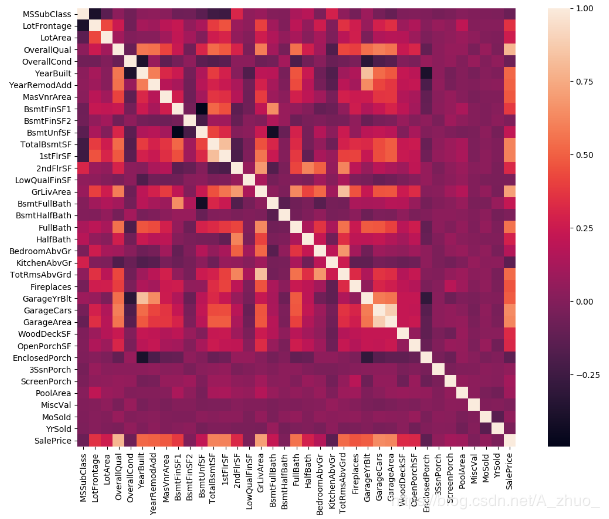
trainData\_exceptID = trainData.drop(['Id'],axis=1)# 剔除ID列

corr = trainData\_exceptID.corr()# 计算变量间的相关系数

plt.figure(figsize=(12,9))

sns.heatmap(corr,square=True) # square设置图是否为正方形

plt.show()

  
根据热力图可以看出：OverallQual（综合质量）、GrLivArea（居住面积）、GarageCars（车库能放多少量车）、GarageArea（车库面积）和SalePrice的相关性最强。其中，GarageCars（车库能放多少量车）和GarageArea（车库面积）两者之间相关性很强，考虑到多重共线性问题，因此可以去掉其中一个变量。

# 3.2 可视化

# 3.2.1 SalePrice随OverallQual变化（箱线图更合适，先用一下散点图看看）

overallQual = trainData\_exceptID['OverallQual']

salePrice = trainData\_exceptID['SalePrice']

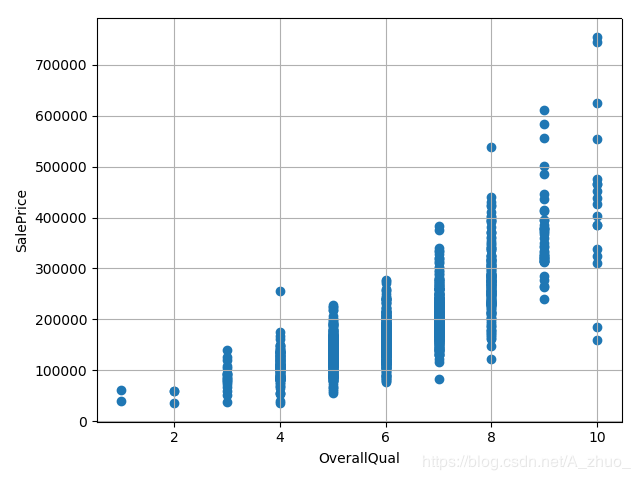
plt.scatter(overallQual,salePrice)

plt.xlabel('OverallQual')

plt.ylabel('SalePrice')

plt.grid()

plt.show()

  
可以用箱线图展示（见下文）

# 3.2.2 SalePrice随GrLivArea（居住面积）变化的散点图

gArea = trainData\_exceptID['GrLivArea']

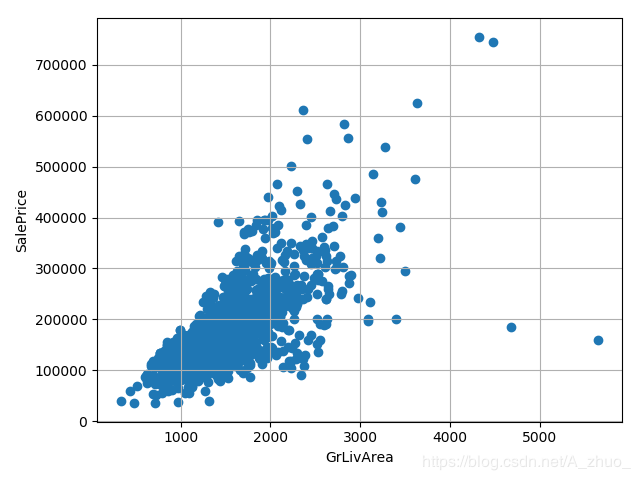
plt.scatter(gArea,salePrice)

plt.xlabel('GrLivArea')

plt.ylabel('SalePrice')

plt.grid()

plt.show()

  
随着居住面积越大，售价越高，很明显，图中存在两个异常点。  
我们再看一个GarageArea（车库面积）和房价间关系：

# 3.2.3 SalePrice随GarageArea（车库面积）变化的散点图

ggArea = trainData\_exceptID['GarageArea']

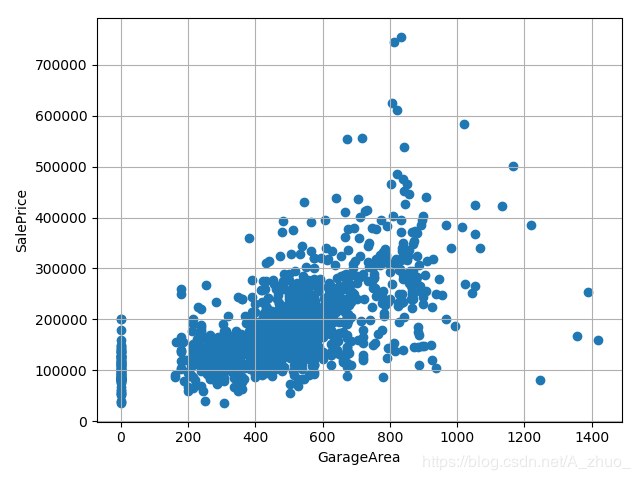
plt.scatter(ggArea,salePrice)

plt.xlabel('GarageArea')

plt.ylabel('SalePrice')

plt.grid()

plt.show()

  
随着车库面积的增大，房价越高。同时似乎也存在异常值。  
接下来画离散变量的箱线图：

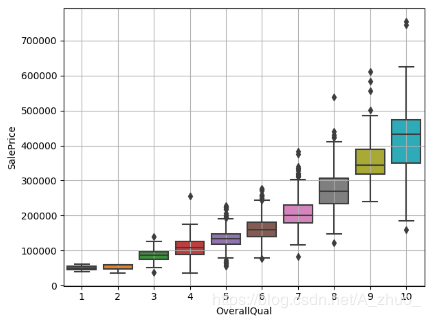
# 3.2.4 SalePrice随OverallQual变化，每一个特征按照箱线图展示

price\_qual = pd.concat([trainData\_exceptID['SalePrice'],trainData\_exceptID['OverallQual']],axis=1) # axis=1按照列维度拼接，行数不变

sns.boxplot(x='OverallQual',y='SalePrice',data=price\_qual)

plt.grid()

plt.show()

  
同散点图趋势一样，随着综合质量的上升，房价也随之上升。