葡萄酒质量预测

1. 导入数据

导入所需的Python库并加载白葡萄酒和红葡萄酒的两个数据集。

```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
from matplotlib import font_manager

wine_red = pd.read_csv('winequality-red.csv', sep=';')
wine_while = pd.read_csv('winequality-white.csv', sep=';')
wine = pd.concat([wine_red, wine_while], axis=0)
wine.head()
```

In [1]:
```
'''导入所需的Python库并加载白葡萄酒和红葡萄酒的两个数据集。'''
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns

wine_red = pd.read_csv('winequality-red.csv', sep=';')
wine_while = pd.read_csv('winequality-white.csv', sep=';')
wine = pd.concat([wine_red, wine_while], axis=0)
wine.head()
```

Out[1]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

葡萄酒成分列表：

列名	含义
fixed acidity	挥发性酸
volatile acidity	挥发性酸
citric acid	柠檬酸
residual sugar	剩余糖分
chlorides	氯化物
free sulfur dioxide	游离二氧化硫
total sulfur dioxide	总二氧化硫
density	密度
pH	酸碱度
sulphates	硫酸盐
alcohol	酒精
quality	质量

2. 检查是数据否有空值

检查是否有空值

```
wine.isnull().sum()
```

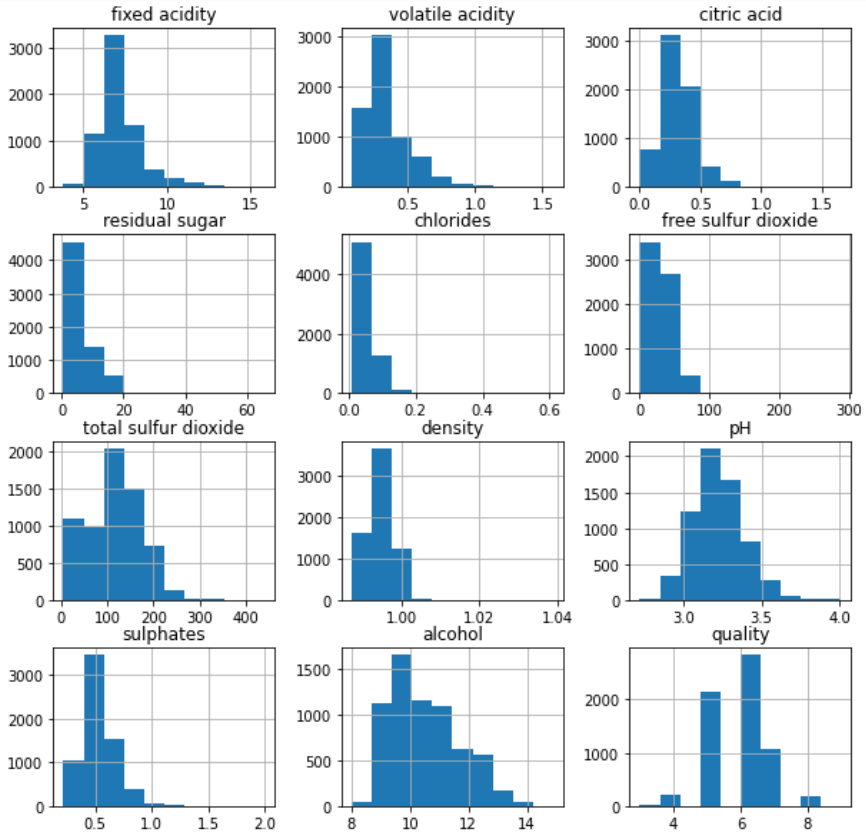
```
In  [2]: '''检查是否有空值'''
         wine.isnull().sum()

Out[2]: '检查是否有空值'

Out[2]: fixed acidity           0
        volatile acidity        0
        citric acid             0
        residual sugar          0
        chlorides               0
        free sulfur dioxide     0
        total sulfur dioxide    0
        density                 0
        pH                      0
        sulphates               0
        alcohol                 0
        quality                 0
        dtype: int64
```

检查发现数据集无空值，则不必进行空值处理

3. 绘制直方图查看数据的分布

查看各个特征值的分布数量

```
wine.hist(figsize=(10, 10))
```

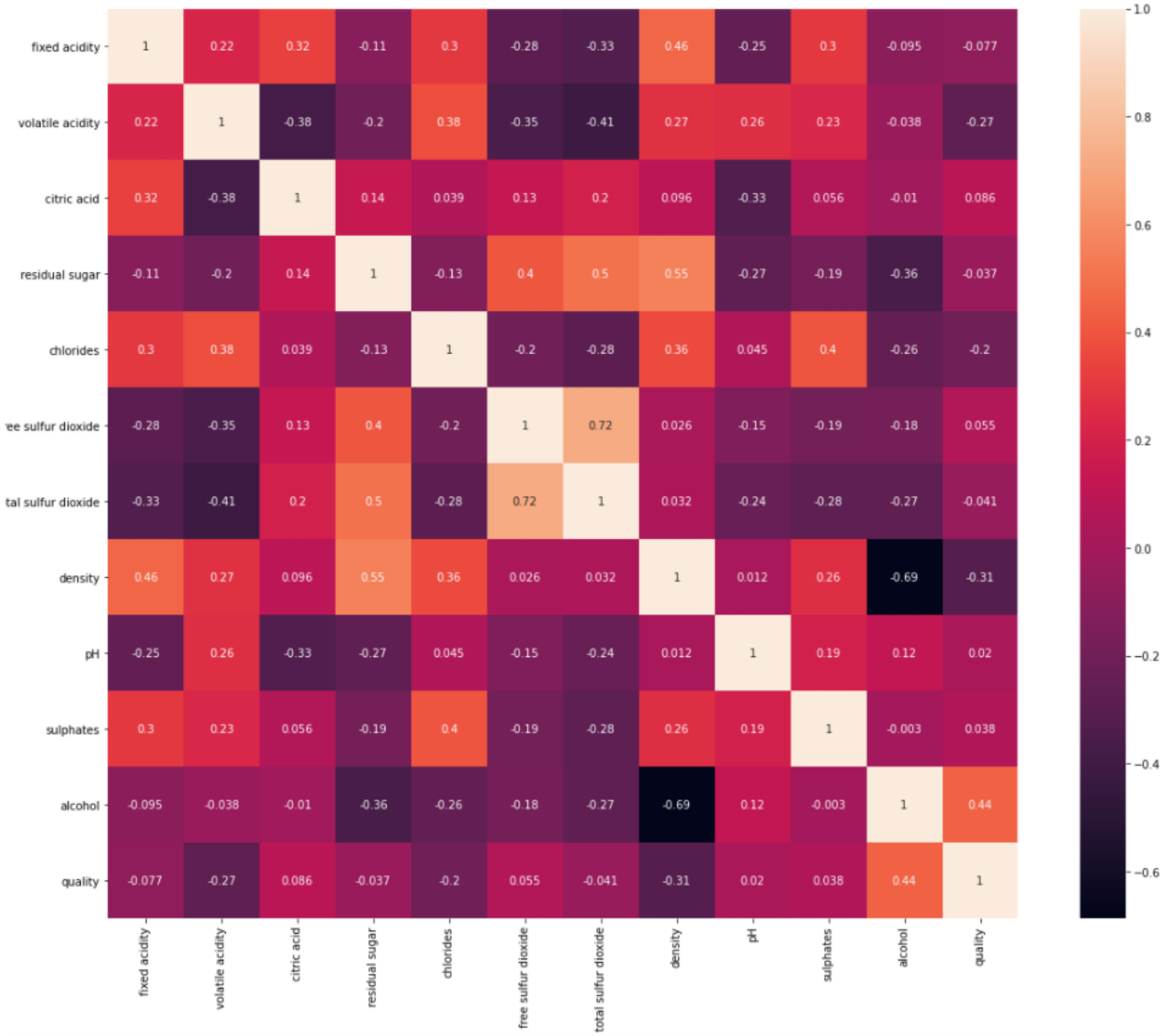


横坐标为特征值，纵坐标为数量

4. 绘制各个特征量与葡萄酒质量的相关性

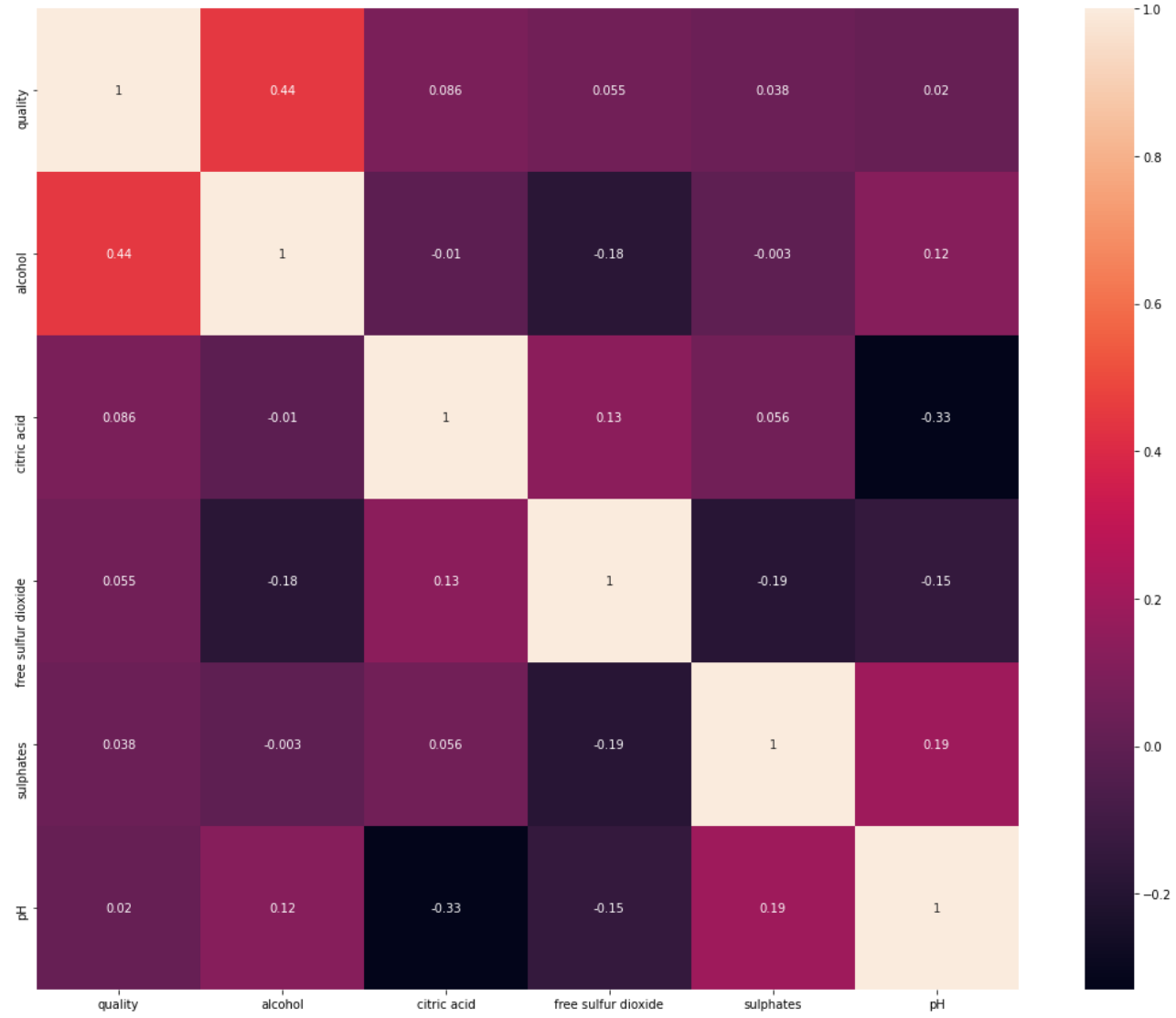
找出输出(质量)变量与所有输入变量之间的相关性：

```
plt.subplots(figsize=(20,15))    #定义热力图大小为（20，15）
corr = wine.corr()    #相关系数
sns.heatmap(corr,square=True, annot=True)    #以corr为数据创建热力图
```



图中发现citric acid(柠檬酸)、free sulfur dioxide(游离二氧化硫)、pH(酸碱度)、sulphates(硫酸盐)、alcohol(酒精)与quality(质量)为正相关，于是我们选择这六个特征量绘制相关性图

```
cols = corr.nlargest(6, 'quality')['quality'].index    #找到与目标值相关性最大的6个特征，而这几个特征之间的相关性要低。
corrcoef = np.corrcoef(wine[cols].values.T)
plt.subplots(figsize=(20,15))    #设置热力图大小
sns.heatmap(corrcoef,square=True, annot=True, xticklabels= cols.values, yticklabels=cols.values)    #创建热力图
```



5. 用各种算法计算拟合训练数据，并根据测试值确定预测输出的准确性

准备工作

使用机器学习中的sklearn库，将数据集拆分为测试和训练数据集，我使用了20%的数据作为测试数据集：

```
y = wine["quality"]    #获取quality列
x = wine.drop(columns=["quality", "fixed acidity", "volatile acidity", "residual sugar", "chlorides",
                       "total sulfur dioxide", "density"], axis=1)    #删去quality列和负相关的列
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
```

由于不同的列具有不同的值，因此使用StandardScaler库归一化值以获得准确的预测结果

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.fit_transform(x_test)
```

一、线性回归

```
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg.fit(x_train, y_train)
pred_logreg = logreg.predict(x_test)    #逻辑回归预测葡萄酒的质量
accuracy = accuracy_score(pred_logreg, y_test)    #计算分类准确率分数
print("Logreg Accuracy Score %.2f" % accuracy)
```

结果图:

```
Out[8]: '现在，我将根据各种算法拟合我的训练数据，并根据测试值确定预测输出的准确性。Python实现如下：'
Out[8]: '线性算法：'
Out[8]: LogisticRegression()
        Logreg Accuracy Score 0.49
```

二、KNN

```
from sklearn.neighbors import KNeighborsClassifier
cm = confusion_matrix(pred_logreg, y_test)    #混淆矩阵
knn = KNeighborsClassifier(n_neighbors=1)    #K近邻算法，选取最近的点的个数为1
knn.fit(x_train, y_train)
pred_knn = knn.predict(x_test)    #knn算法预测葡萄酒的质量
accuracy = accuracy_score(pred_knn, y_test)    #计算分类准确率分数
print("Knn Accuracy Score %.2f" % accuracy)
```

结果图:

```
Out[9]: 'KNN'
Out[9]: KNeighborsClassifier(n_neighbors=1)
        Knn Accuracy Score 0.59
```

三、支持向量机SVC

```
from sklearn.svm import SVC
svc = SVC()
svc.fit(x_train, y_train)    #训练数据集
pred_svc =svc.predict(x_test)    #SVC预测葡萄酒的质量
accuracy = accuracy_score(pred_svc, y_test)    #计算分类准确率分数
print("SVC Accuracy Score %.2f" % accuracy)
```

结果图:

```
Out[10]: '支持向量机SVC'
Out[10]: SVC()
         SVC Accuracy Score 0.52
```

四、随机森林

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf.fit(x_train, y_train)    #训练数据集
pred_rf =rf.predict(x_test)    #随机森林预测葡萄酒的质量
accuracy = accuracy_score(pred_rf, y_test)    #计算分类准确率分数
print("Random Forest Accuracy Score %.2f" % accuracy)
```

结果图:

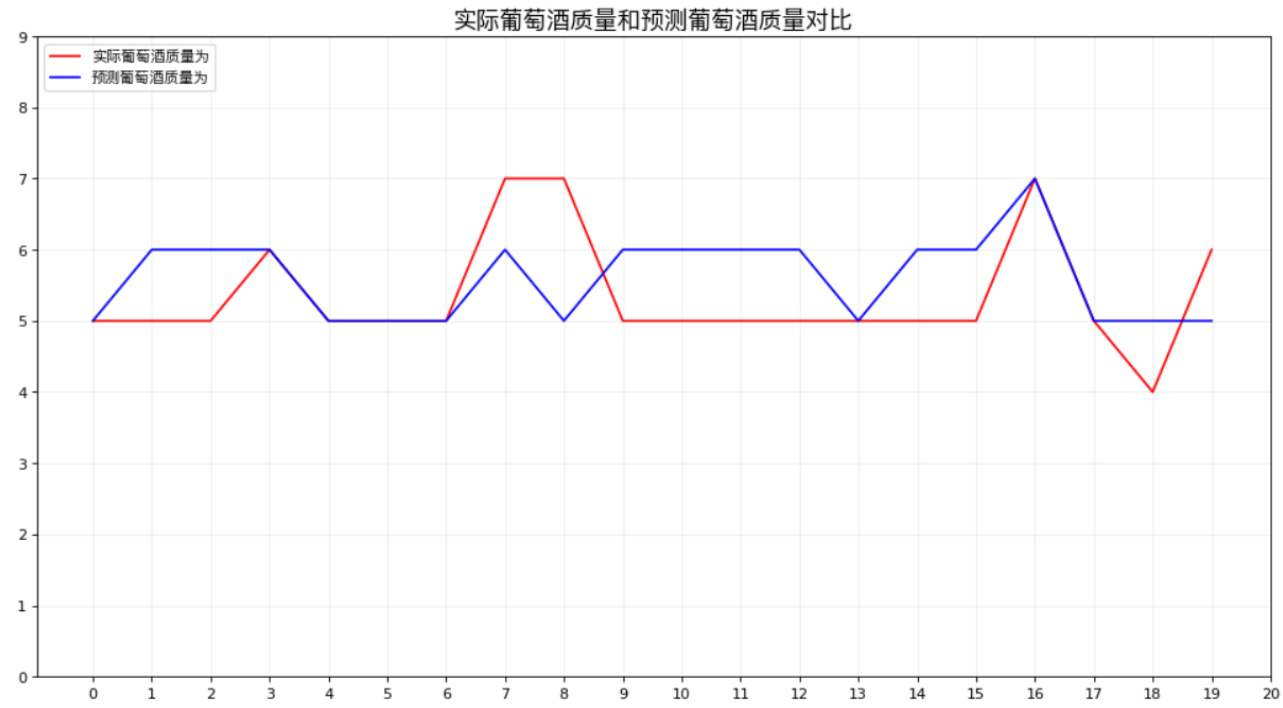
```
Out[11]: '随机森林'

Out[11]: RandomForestClassifier()

         Random Forest Accuracy Score 0.66
```

在以上四种算法中，我们发现随机森林算法的预测准确率最高，所以接下来我们选用随机森林算法来比较实际葡萄酒质量和预测葡萄酒质量。

我们从x_test测试数据集中徐州呢前20行创建新的观测集

```python
new_observations = x.head(20)  # 用数据集中的前20行创建新观测集
new_observations = scaler.fit_transform(new_observations)  #对new_observations数据进行归一化
y_predicted = rf.predict(new_observations)  # 用随机森林法对新观测集进行质量预测

#设置字体
my_font = font_manager.FontProperties(fname=r"C:/Users/xjhqre/PingFang SC.ttc")
#设置图片大小
plt.figure(figsize=(15, 8), dpi=80)
#画图—折线图
plt.plot(range(len(y.head(20).values)), y.head(20).values, label='实际葡萄酒质量为',color="r")
plt.plot(range(len(y_predicted)), y_predicted, label='预测葡萄酒质量为',color="b")
#设置x,y坐标
plt.xticks(range(len(y_predicted)+1))
plt.yticks(range(10))
#设置网格线
plt.grid(alpha=0.2)
plt.legend(prop=my_font,loc="upper left")
plt.title('实际葡萄酒质量和预测葡萄酒质量对比',fontproperties=my_font, size = 16)
plt.show()
```

结果图：



从折线图上我们可以看到，预测结果和实际结果有些许不同，但总体上还是一致的