西安交通大学实验报告

课程 **Python数据处理** 实验名称 **使用scikit-learn构建模型**  共 **16**页

系 别 **电信学部** 实 验 日 期 **2023**年**4**月**26**日

专业班级 实 验 报 告 日 期 **2023**年**4**月**26**日

姓 名 学号 报 告 退 发( 订正 、 重做 )

同 组 人\_\_\_无\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 教 师 审 批 签 字

*本次实验合计约使用****3****个小时*

# **实验目的**

1.掌握使用scikit-learn构建线性模型和梯度提升回归模型的技能  
2.熟练划分数据集为训练集和测试集的方法  
3.学会将数据集分离为数据和标签  
4.熟练降维数据集并构建聚类模型，同时了解如何对其进行评价

# **实验内容**

## 实验题目一：（35分）

**使用sklearn完成以下实验题目，实验题目所用数据集从思源学堂中下载。**

**1）使用sklearn读取数据集wine**

**2）拆分数据集wine的数据和标签（class）**

**3）对数据集wine进行标准化**

**4）对数据集wine进行PCA降维**

**5）构建聚类数目为3的K-Means模型**

**6）使用FMI评价聚类模型**

**7）确定最佳聚类数目（2~10类）**

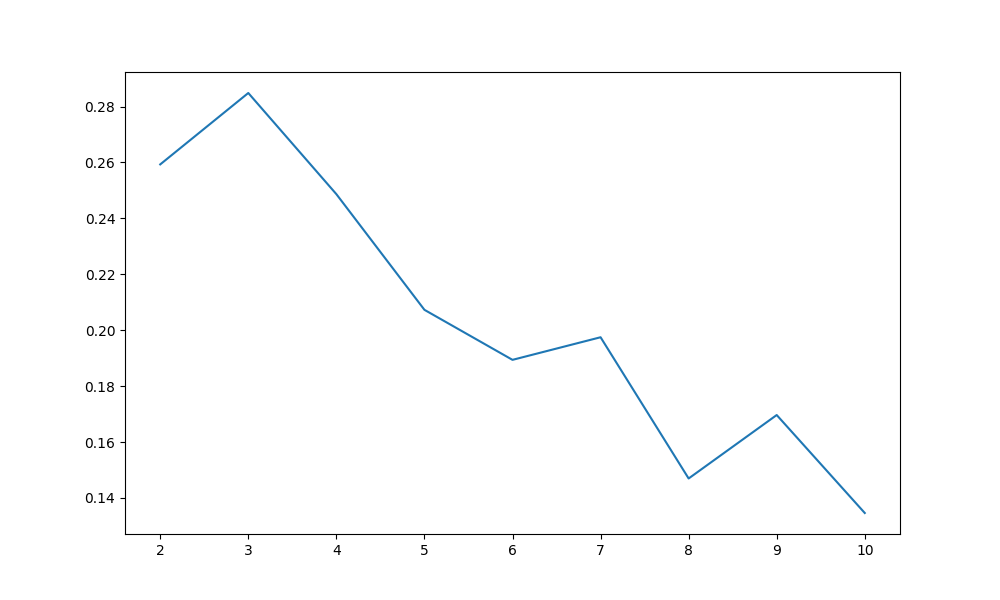
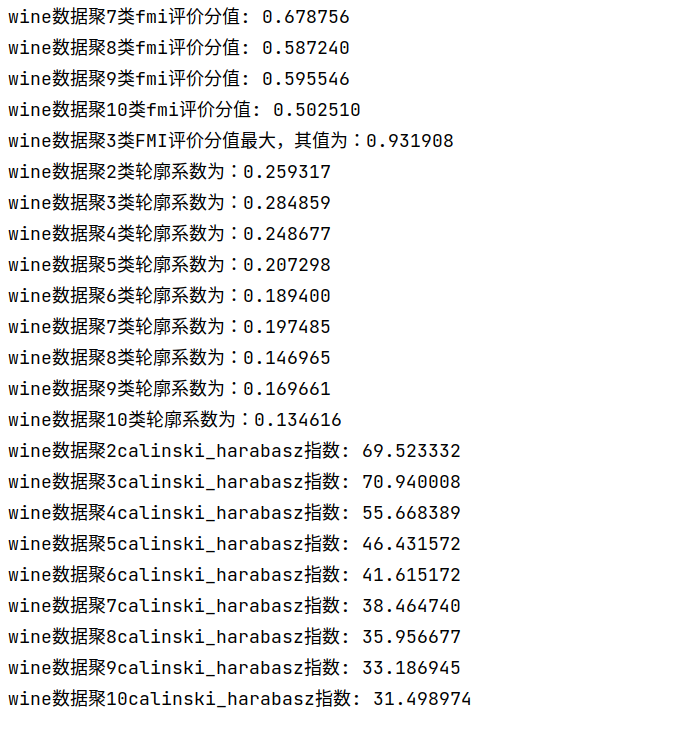
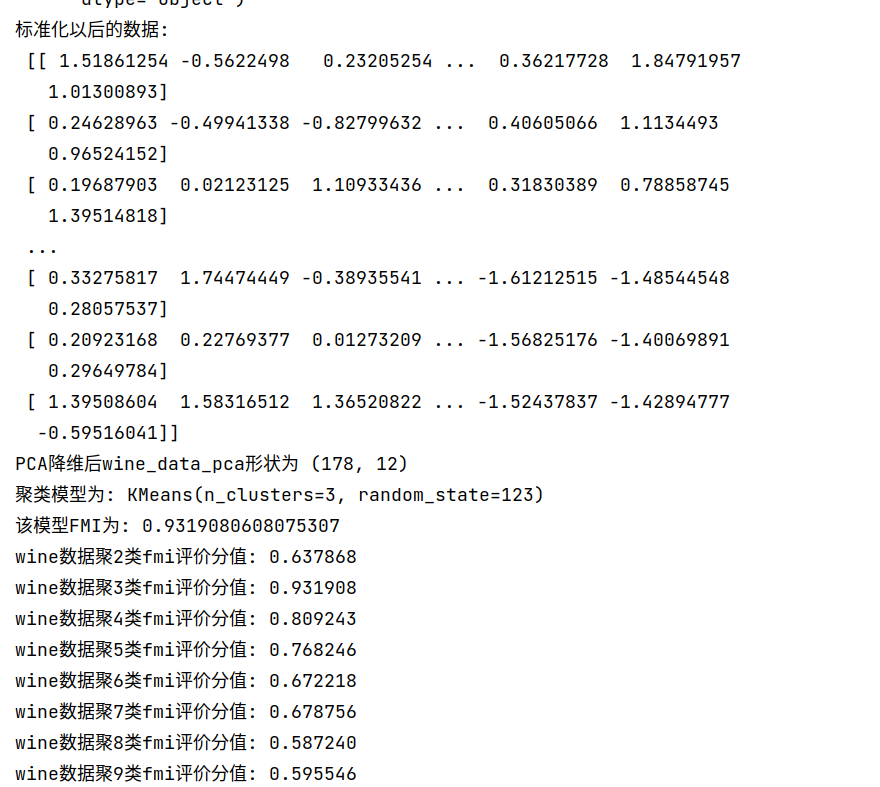
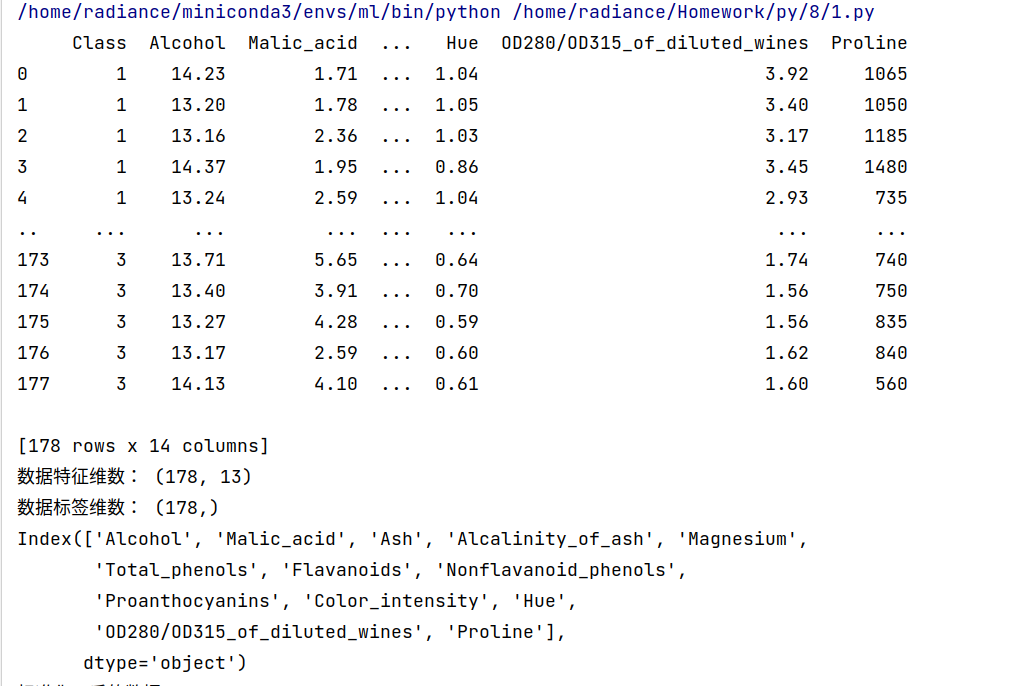
**8）使用轮廓系数评价聚类模型**

**9）使用Calinski-Harabasz指数评价聚类模型**

#### 程序源代码：

import pandas as pd  
from sklearn.preprocessing import StandardScaler  
from sklearn.decomposition import PCA  
from sklearn.cluster import KMeans  
from sklearn.metrics import fowlkes\_mallows\_score, silhouette\_score, calinski\_harabasz\_score  
import matplotlib.pyplot as plt  
  
*# 读取数据*wine = pd.read\_csv('wine.csv', sep=',', encoding='utf-8')  
print(wine)  
  
*# 拆分数据集wine的标签和数据*wine\_data = wine.iloc[:, 1:].values  
wine\_class = wine.iloc[:, 0].values  
wine\_names = wine.columns[1:]  
print('数据特征维数：', wine\_data.shape)  
print('数据标签维数：', wine\_class.shape)  
print(wine\_names)  
  
*# 对数据集wine进行标准化*stdScale = StandardScaler().fit(wine\_data)  
wine\_data\_scale = stdScale.transform(wine\_data)  
print('标准化以后的数据:\n', wine\_data\_scale)  
  
*# 对数据集wine进行PCA降维*wine\_data\_pca = PCA(n\_components='mle', random\_state=123).fit\_transform(wine\_data)  
print('PCA降维后wine\_data\_pca形状为', wine\_data\_pca.shape)  
  
*# 构建并训练模型  
  
# 构建聚类数目为三的K-Means模型*kmeans = KMeans(n\_clusters=3, random\_state=123).fit(wine\_data\_scale)  
print('聚类模型为:', kmeans)  
  
*# 使用FMI评价聚类模型*score = fowlkes\_mallows\_score(wine\_class, kmeans.labels\_)  
print('该模型FMI为:', score)  
  
*# 确定最佳聚类数目(2-10)*max\_i = 0  
max\_score = 0  
for i in range(2, 11):  
 kmeans = KMeans(n\_clusters=i, random\_state=123).fit(wine\_data\_scale)  
 score = fowlkes\_mallows\_score(wine\_class, kmeans.labels\_)  
 print('wine数据聚%d类fmi评价分值: %f' % (i, score))  
 if score > max\_score:  
 max\_i = i  
 max\_score = score  
print('wine数据聚%d类FMI评价分值最大，其值为：%f' % (max\_i, max\_score))  
  
*# 使用轮廓系数评价聚类模型*silhouetteScore = []  
for i in range(2, 11):  
 kmeans = KMeans(n\_clusters=i, random\_state=123).fit(wine\_data\_scale)  
 score = silhouette\_score(wine\_data\_scale, kmeans.labels\_)  
 silhouetteScore.append(score)  
 print('wine数据聚%d类轮廓系数为：%f' % (i, score))  
plt.figure(figsize=(10, 6))  
plt.plot(range(2, 11), silhouetteScore, linewidth=1.5, linestyle='-')  
plt.show()  
  
*# 使用Calinki-harabasz指数评价聚类模型*for i in range(2, 11):  
 kmeans = KMeans(n\_clusters=i, random\_state=123).fit(wine\_data\_scale)  
 score = calinski\_harabasz\_score(wine\_data\_scale, kmeans.labels\_)  
 print('wine数据聚%dcalinski\_harabasz指数: %f' % (i, score))

#### 运行结果：

**?**

## 实验题目二：（30分）

**使用sklearn完成以下实验题目，实验题目所用数据集从思源学堂中下载。**

**1）使用sklearn读取数据集wine**

**2）拆分数据集wine的数据和标签（class）**

**3）将数据集wine划分为训练集和测试集**

**4）使用离差标准化标准化数据集**

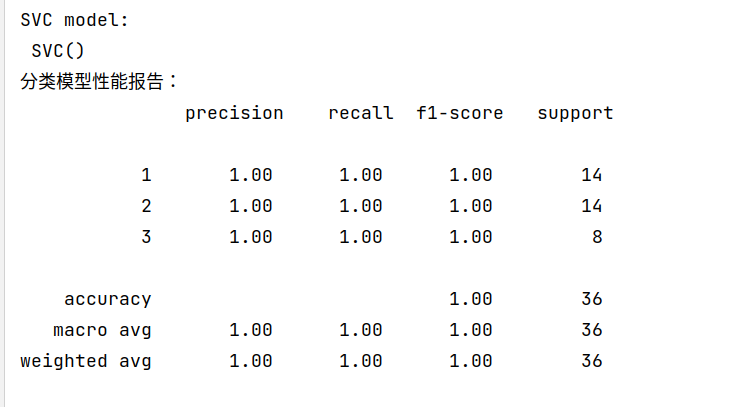
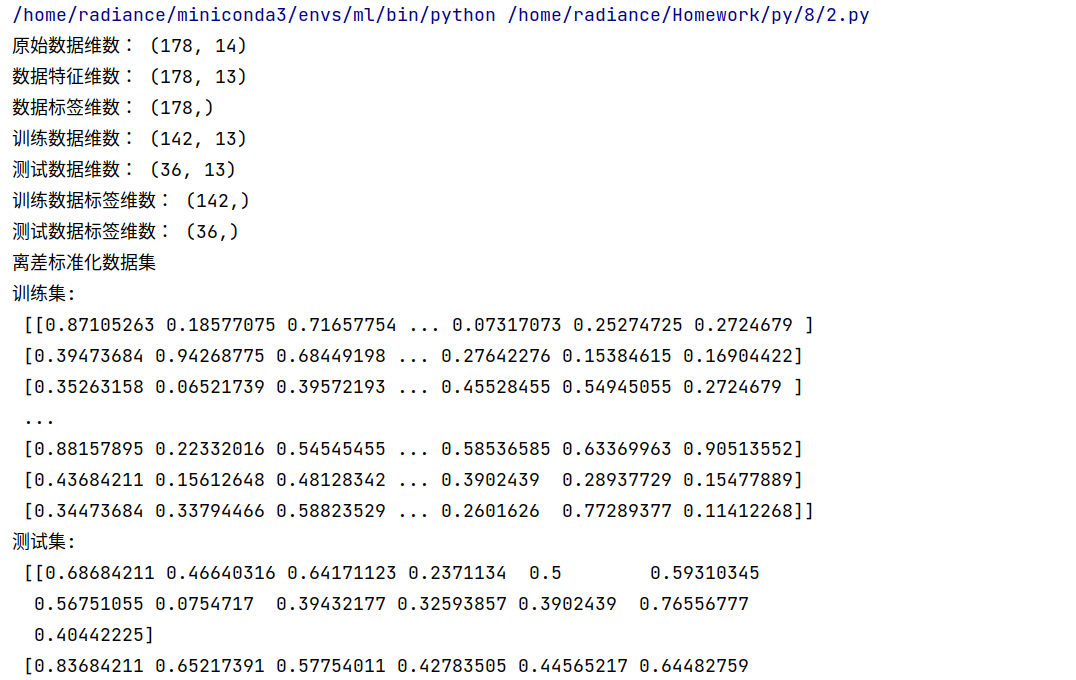
**5）构建SVM模型**

**6）给出评价分类模型性能的分类报告**

#### 程序源代码

import pandas as pd  
from sklearn.model\_selection import train\_test\_split  
from sklearn.preprocessing import MinMaxScaler  
from sklearn.svm import SVC  
from sklearn.metrics import classification\_report  
  
*# 读取数据*wine = pd.read\_csv('wine.csv', sep=',', encoding='utf-8')  
print('原始数据维数：', wine.shape)  
  
*# 拆分数据和标签*wine\_data = wine.iloc[:, 1:].values  
wine\_class = wine.iloc[:, 0].values  
print('数据特征维数：', wine\_data.shape)  
print('数据标签维数：', wine\_class.shape)  
  
*# 将数据集划分为训练集和测试集*data\_train, data\_test, class\_train, class\_test = train\_test\_split(  
 wine\_data, wine\_class, test\_size=0.2, random\_state=42)  
print('训练数据维数：', data\_train.shape)  
print('测试数据维数：', data\_test.shape)  
print('训练数据标签维数：', class\_train.shape)  
print('测试数据标签维数：', class\_test.shape)  
  
*# 离差标准化数据集*print("离差标准化数据集")  
scale = MinMaxScaler().fit(wine\_data)  
data\_train\_scale = scale.transform(data\_train)  
data\_test\_scale = scale.transform(data\_test)  
print('训练集:\n', data\_train\_scale)  
print('测试集:\n', data\_test\_scale)  
  
*# 构建SVM模型，训练模型*wine\_svc = SVC().fit(data\_train\_scale, class\_train)  
print('SVC model:\n', wine\_svc)  
  
*# 输出模型评估报告*class\_pred = wine\_svc.predict(data\_test\_scale)  
report = classification\_report(class\_test, class\_pred)  
print('分类模型性能报告：\n', report)

#### 运行结果：



## 实验题目三：（35分）

**使用sklearn完成以下实验题目，实验题目所用数据集从思源学堂中下载。**

**1）使用sklearn读取数据集winequality**

**2）拆分数据集winequality的数据和标签（quality）**

**3）将数据集winequality划分为训练集和测试集**

**4）构建线性回归模型**

**5）计算线性回归模型的平均绝对误差、均方误差、中值绝对误差、可解释方差和R^2**

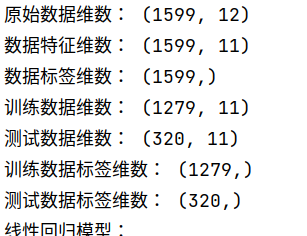
**6）构建梯度提升回归模型**

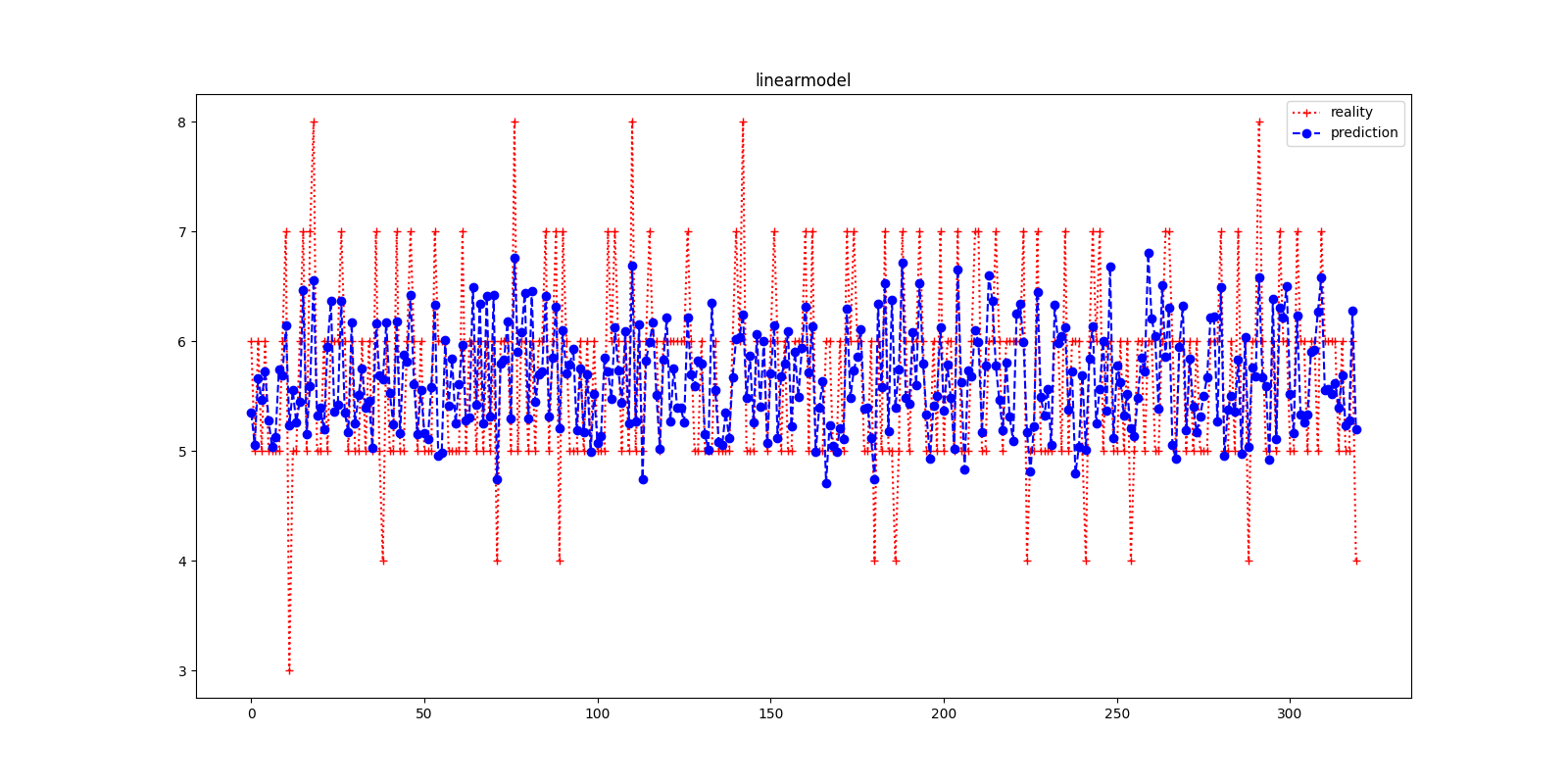
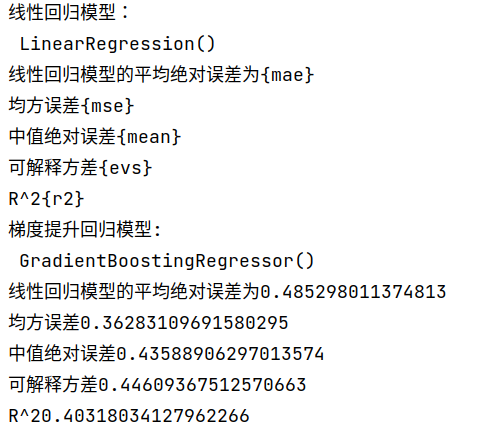
**7）计算梯度提升回归模型的平均绝对误差、均方误差、中值绝对误差、可解释方差和R^2**

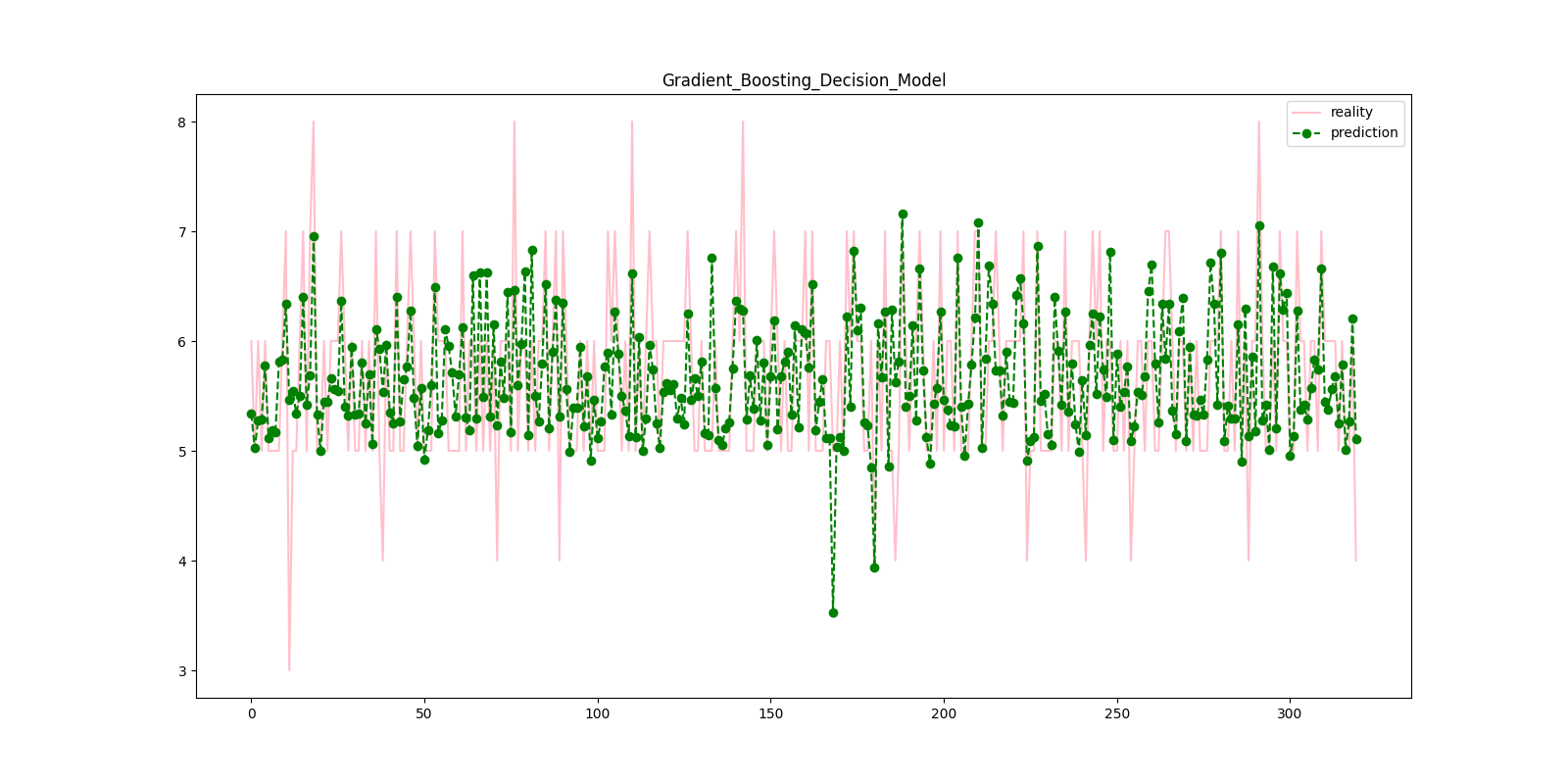
#### 程序源代码：

*# 读取数据集*import pandas as pd  
  
wine = pd.read\_csv('winequality.csv', sep=';', encoding='utf-8')  
print('原始数据维数：', wine.shape)  
  
*# 拆分数据集的数据和标签  
# print(wine.columns)*wine\_data = wine.iloc[:, 0:-1]  
wine\_quality = wine.iloc[:, -1]  
  
print('数据特征维数：', wine\_data.shape)  
print('数据标签维数：', wine\_quality.shape)  
*# 将数据集划分为训练集和测试集*from sklearn.model\_selection import train\_test\_split as split  
  
wine\_data\_train, wine\_data\_test, wine\_quality\_train, wine\_quality\_test = split(wine\_data, wine\_quality, test\_size=0.2,  
 random\_state=42)  
  
print('训练数据维数：', wine\_data\_train.shape)  
print('测试数据维数：', wine\_data\_test.shape)  
print('训练数据标签维数：', wine\_quality\_train.shape)  
print('测试数据标签维数：', wine\_quality\_test.shape)  
  
*# 构建线性回归模型*from sklearn.linear\_model import LinearRegression  
  
quality\_linear = LinearRegression().fit(wine\_data\_train, wine\_quality\_train)  
print('线性回归模型：\n', quality\_linear)  
wine\_quality\_pred = quality\_linear.predict(wine\_data\_test)  
import matplotlib.pyplot as plt  
  
plt.figure(figsize=(16, 8))  
plt.title('linearmodel')  
plt.plot(range(len(wine\_quality\_test)), wine\_quality\_test, 'r+:',  
 range(len(wine\_quality\_pred)), wine\_quality\_pred, 'bo--')  
plt.legend(['reality', 'prediction'])  
plt.savefig('linearmodel.png')  
plt.show()  
*# 计算线性回归模型的平均绝对误差、均方误差、中值绝对误差、可解释方差和R^2*from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, median\_absolute\_error, explained\_variance\_score, \  
 r2\_score  
  
wine\_quality\_pred = quality\_linear.predict(wine\_data\_test) *# 在上一题中预测出的值*mae = mean\_absolute\_error(wine\_quality\_test, wine\_quality\_pred)  
mse = mean\_squared\_error(wine\_quality\_test, wine\_quality\_pred)  
mean = median\_absolute\_error(wine\_quality\_test, wine\_quality\_pred)  
evs = explained\_variance\_score(wine\_quality\_test, wine\_quality\_pred)  
r2 = r2\_score(wine\_quality\_test, wine\_quality\_pred)  
print("线性回归模型的平均绝对误差为{mae}\n均方误差{mse}\n中值绝对误差{mean}\n可解释方差{evs}\nR^2{r2}")  
*# 构建梯度提升回归模型*import sklearn.ensemble  
from sklearn.ensemble import GradientBoostingRegressor  
  
quality\_gradient = GradientBoostingRegressor().fit(wine\_data\_train, wine\_quality\_train)  
print('梯度提升回归模型:\n', quality\_gradient)  
wine\_quality\_pred2 = quality\_gradient.predict(wine\_data\_test)  
import matplotlib.pyplot as plt  
  
plt.figure(figsize=(16, 8))  
plt.title('Gradient\_Boosting\_Decision\_Model')  
plt.plot(range(len(wine\_quality\_test)), wine\_quality\_test, 'pink+',  
 range(len(wine\_quality\_pred2)), wine\_quality\_pred2, 'go--')  
plt.legend(['reality', 'prediction'])  
plt.savefig('Boost.png')  
plt.show()  
*# 计算梯度提升回归模型的平均绝对误差、均方误差、中值绝对误差、可解释方差和R^2*from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, median\_absolute\_error, explained\_variance\_score, \  
 r2\_score  
  
wine\_quality\_pred2 = quality\_gradient.predict(wine\_data\_test) *# 在上一题中预测出的值*mae1 = mean\_absolute\_error(wine\_quality\_test, wine\_quality\_pred2)  
mse1 = mean\_squared\_error(wine\_quality\_test, wine\_quality\_pred2)  
mean1 = median\_absolute\_error(wine\_quality\_test, wine\_quality\_pred2)  
evs1 = explained\_variance\_score(wine\_quality\_test, wine\_quality\_pred2)  
r21 = r2\_score(wine\_quality\_test, wine\_quality\_pred)  
print(f"线性回归模型的平均绝对误差为{mae1}\n均方误差{mse1}\n中值绝对误差{mean1}\n可解释方差{evs1}\nR^2{r21}")

#### 运行结果：







# **实验小结**

**本次实验中，我们学会了构建聚类模型时应使用标准化以后的数据的重要性，这样可以避免因为数据范围不同而导致的误差。同时，我们也学会了如何构建svm模型，并利用分类报告对模型性能进行评价。在线性回归模型方面，我们熟练掌握了计算模型的平均绝对误差、均方误差、中值绝对误差、可解释方差和R^2，这些指标对于评估模型的准确性和可靠性非常重要。最后，我们学会了使用 FMI、轮廓系数、Calinski-Harabasz指数等指标来评价聚类模型的性能，并加深了对这些指标的理解。通过本次实验，我们掌握了多种机器学习技术的应用和评价方法，为今后的研究和工作打下了坚实基础。**