|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓 名** |  | | **学号** | 211181510 | **班级** | 计科21-2 |
| **实验名称** | 数据线性回归预测学生成绩 | | | | | |
| **课程名称** | Python语言  程序设计 | | | **实验类型** | | 设计型 |
| **实验要求：** | | | | | | |
| 界面清晰，材料丰富，数据可靠，软件需求分析充分，数据处理加工能力强，软件设计合理，运行无错误。对于研究课题能熟练运用所学理论与专业知识，理论联系实际，全面分析问题并解决问题；具有一定的创新性。在实现过程要具有较高的科学素养、学习态度、纪律表现强。语句通顺、流畅，叙述简明扼要，思路层次清晰，概括全面准确，重点突出。 | | | | | | |
|  |
|  |
|  |
| **序号** | **评分项** | | | **满分(100分)** | **得分** | **备注** |  |
| **一、** | **界面设计** | | | **15** |  |  |  |
| **二、** | **算法结构** | | | **25** |  |  |  |
| **三、** | **运行状况** | | | **25** |  |  |  |
| **四、** | **实现功能** | | | **25** |  |  |  |
| **五、** | **文档材料** | | | **10** |  |  |  |
| **总分** |  | | | | | |  |
| **任课教师** | | | | | | |  |
| **上交日期：** | | **2024年6月26日** | | **审核日期：** | | **2024-6-20** |  |

**Python课程设计评分表**

**Python语言**

**课程设计报告**

**设计题目： 数据线性回归预测学生成绩**

**学 院： 网络安全与信息技术学院**

**专 业： 计算机科学与技术**

**学 号： 211181510**

**姓 名：**

**指导教师：**

**2024 年 6 月 26 日**

**目录**

[**1.** **设计目的和内容** 3](#_Toc170347024)

[**1.1设计目的** 3](#_Toc170347025)

[**1.2设计内容** 3](#_Toc170347026)

[**2.** **基本功能描述** 3](#_Toc170347027)

[**实验数据及分析** 4](#_Toc170347028)

[**3.1实验数据背景及位置** 4](#_Toc170347029)

[**3.2数据集格式** 4](#_Toc170347030)

[**3.3 实验分析** 5](#_Toc170347031)

[**3.** **实验设计** 5](#_Toc170347032)

[**3.1 导入库** 5](#_Toc170347033)

[**3.2 导入并初始化数据** 5](#_Toc170347034)

[**3.3 数据处理** 6](#_Toc170347035)

[**4** **结论** 16](#_Toc170347036)

[**5. 附录** 17](#_Toc170347037)

[**5.1实验代码** 17](#_Toc170347038)

[**5.2数据集的属性介绍** 22](#_Toc170347039)

1. **设计目的和内容**

**1.1设计目的**

线性回归是一种统计学方法，用于建立一个或多个自变量（解释变量）与因变量（响应变量）之间的线性关系。它是一种预测分析技术，用于预测连续的数值型数据。

现需使用此方法通过训练集训练和测试集测试学校学生的信息属性来生成多个线性模型，将所给的信息进行数据处理，预测学生成绩**。**

**1.2设计内容**

借助python语言相关的numpy、pandas等库将所拥有的数据集进行处理，直观清晰的为我们反映出可视化数据。分析预测事件

1. **基本功能描述**

主要功能包括数据可视化、统计分析、特征工程、模型训练和评估。

2.1导入必要的库：代码开始部分导入了用于数据处理、可视化、机器学习建模和评估的多个Python库。

2．2数据加载：使用pandas库从CSV文件加载数据集到DataFrame对象。

2.3数据可视化：

·绘制成绩分布的柱状图和箱型图，以展示不同成绩的学生数量。

·绘制年龄分布的密度图和柱状图，以展示不同年龄段的学生人数。

·绘制城乡学生分布图，以展示城市和农村学生的比例。

·绘制成绩与年龄、失败次数的关系图，以分析这些因素对成绩的影响。

2.4数据探索：

·检查数据集中是否存在缺失值。

·分析性别比例、年龄分布等基本统计信息。

2.5特征工程：

·对离散变量进行独热编码，将类别变量转换为数值形式。

·选择与目标变量（G3成绩）相关性最强的特征。

2.6数据集分割：将数据集分割为训练集和测试集，用于模型训练和评估。

2.7基准模型评估：使用中位数作为基准模型，计算其平均绝对误差（MAE）和均方根误差（RMSE）。

2.8模型训练与评估：

·定义一个函数evaluate，用于训练和评估多种回归模型，包括线性回归、ElasticNet回归、随机森林、额外树、支持向量回归、梯度提升回归和基线模型。

·计算每种模型的MAE和RMSE，并存储在DataFrame中。

2.9结果可视化：绘制不同模型的MAE和RMSE的条形图，以比较它们的性能。2.10模型保存：使用pickle库保存训练好的线性回归模型到文件中，以便将来使用。

**实验数据及分析**

**3.1实验数据背景及位置**

现在有一份数据集，该数据接近了学校学生的学习成绩。数据属性包括学生成绩，人口统计学，社会和与学校相关的特征，并通过使用学校报告和调查表进行收集。提供了两个关于两个不同学科表现的数据集：数学（mat）和葡萄牙语（por）

数据集存放在"C:\Users\裕龙\Desktop\机器学习\线性回归预测成绩r\BigDataPredicteGrades-master\student-mat.csv"

**3.2数据集格式**

该数据集共有396条，每列33个属性。

33个属性分别是：学校、性别、年龄、地址-、famsize、Pstatus、Medu、Fedu、Mjob、Fjob、理由、监护人、traveltime、学习时间、失败、schoolup、famsup、付费、、托儿所、更高、互联网-、浪漫、家族、空闲时间、外出、Dalc、Walc、健康、缺勤、G1、G2、G3。如图1学生信息数据集所示：

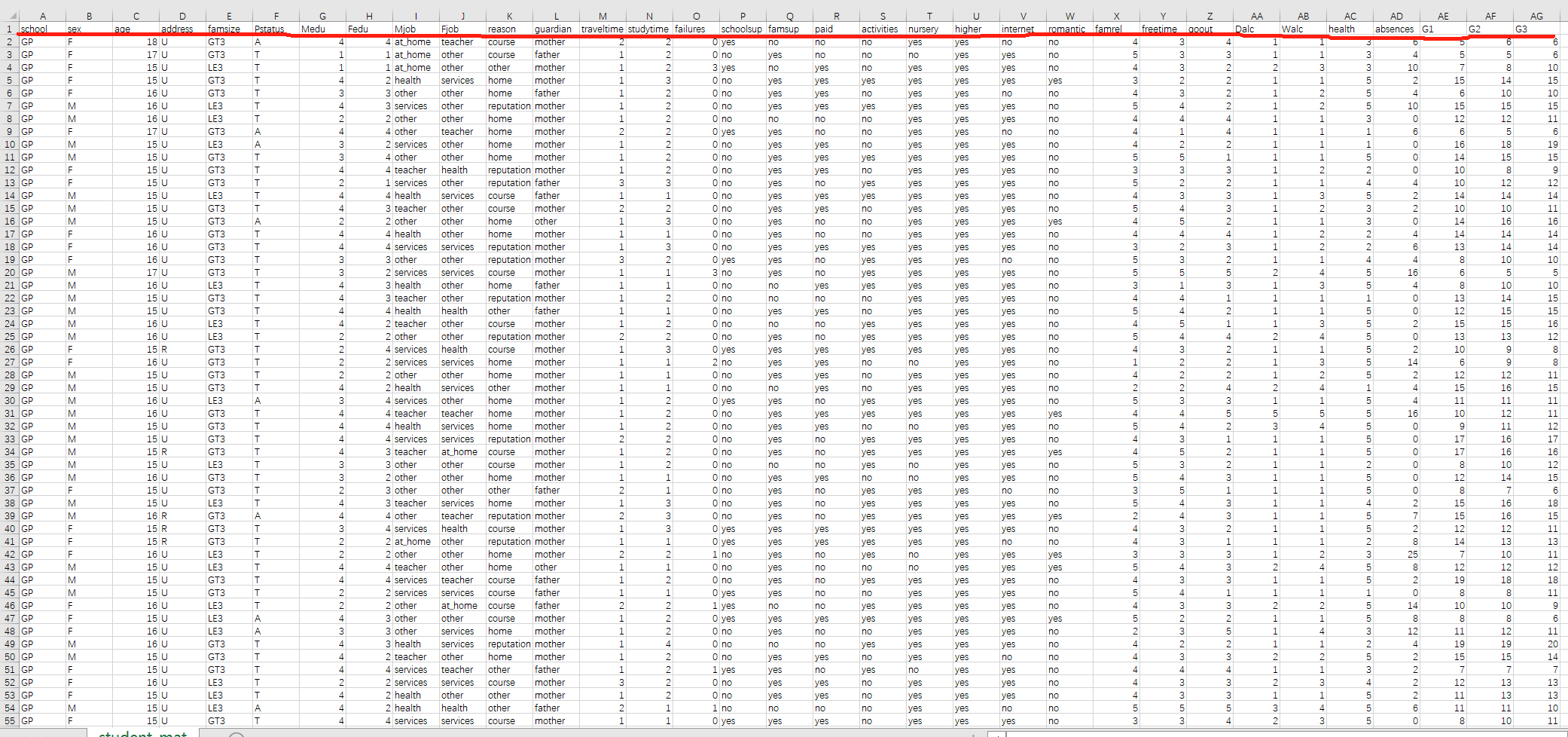


图1 学生信息数据集

**3.3 实验分析**

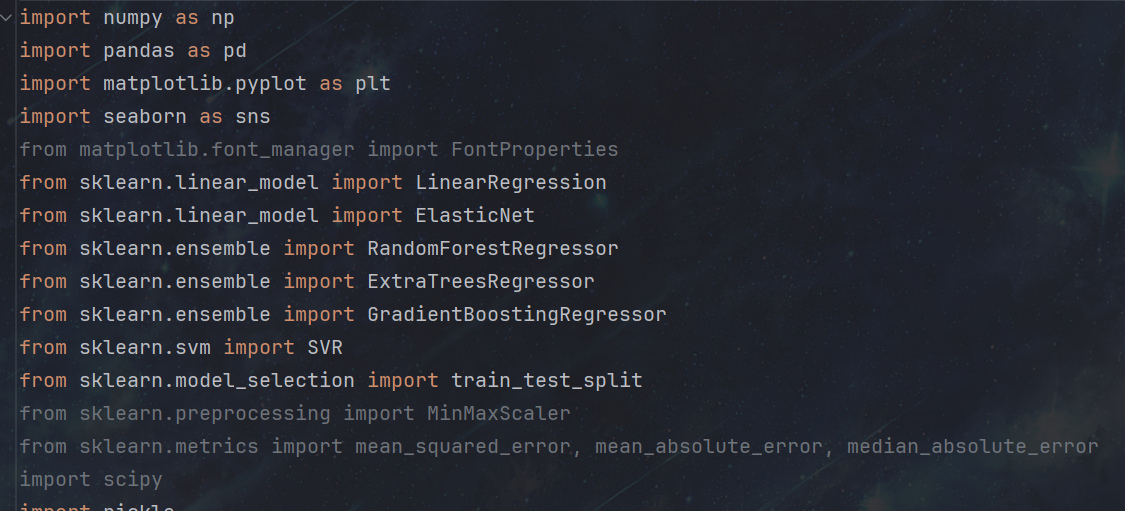
·使用python语言，导入相应的库，导入所需的数据，做数据的预处理

·构建简单线性回归模型

·使用数据集中不同的参数，进行线性回归模型的比较

1. **实验设计**

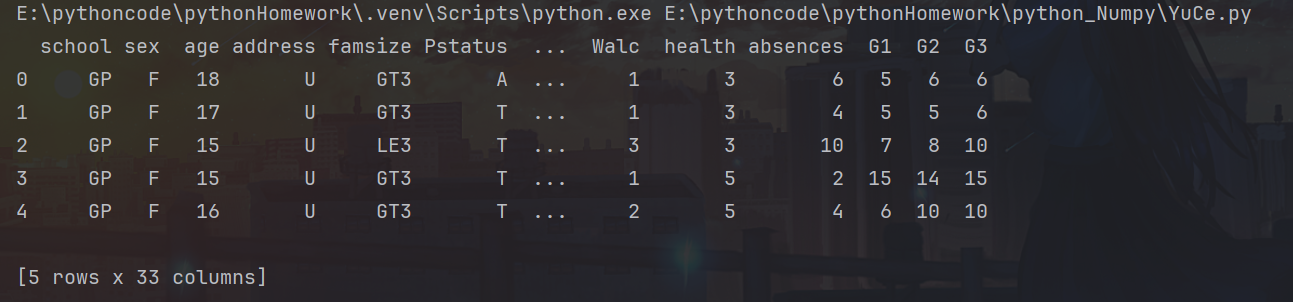
**3.1 导入库**



**3.2 导入并初始化数据**

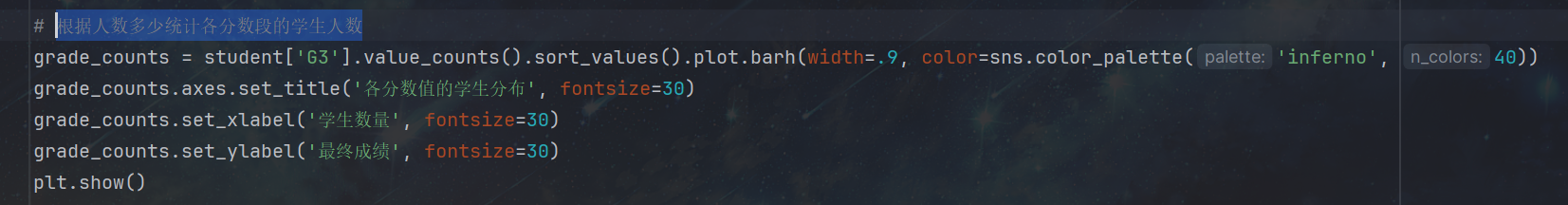


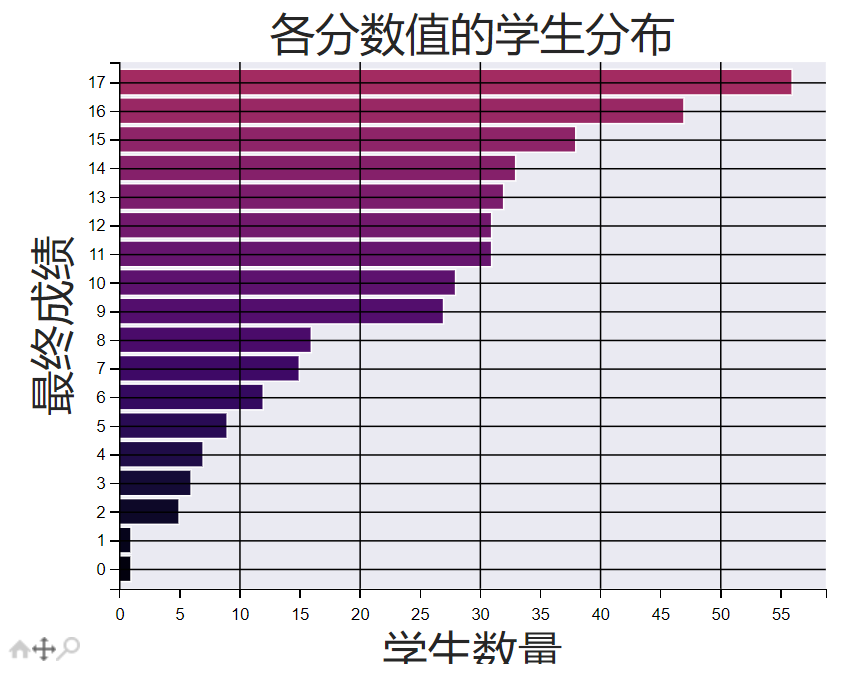
输出结果：



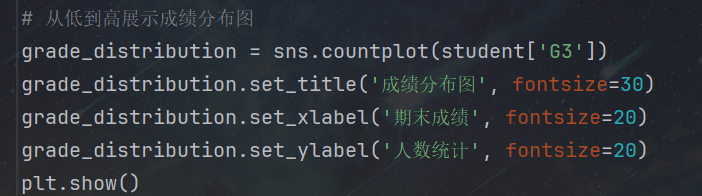
**3.3 数据处理**

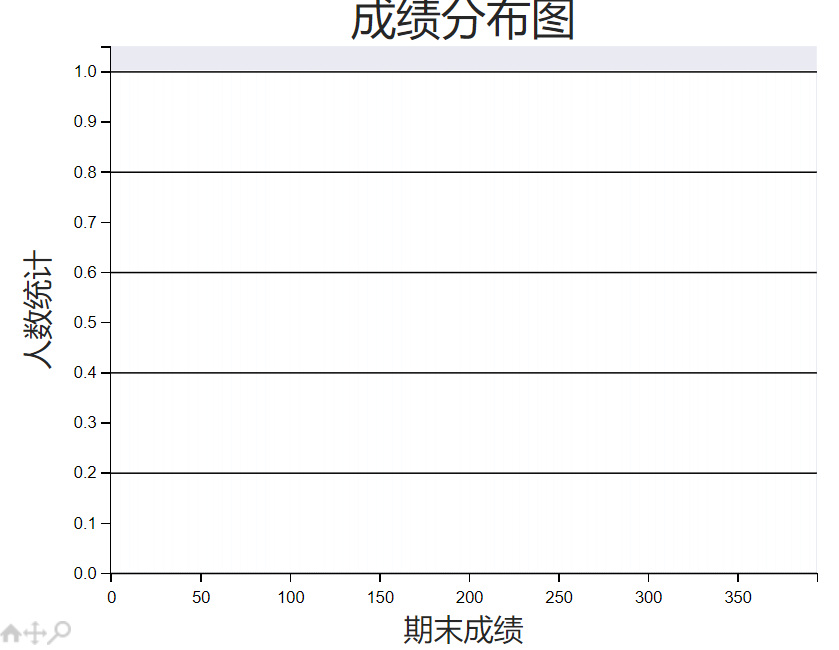
3.3.1 根据人数多少统计各分数段的学生人数



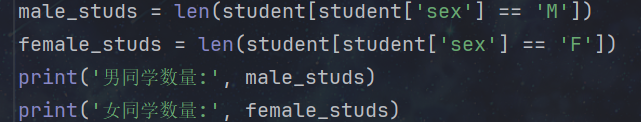


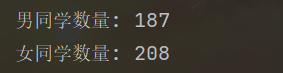
3.3.2 从低到高展示成绩分布图





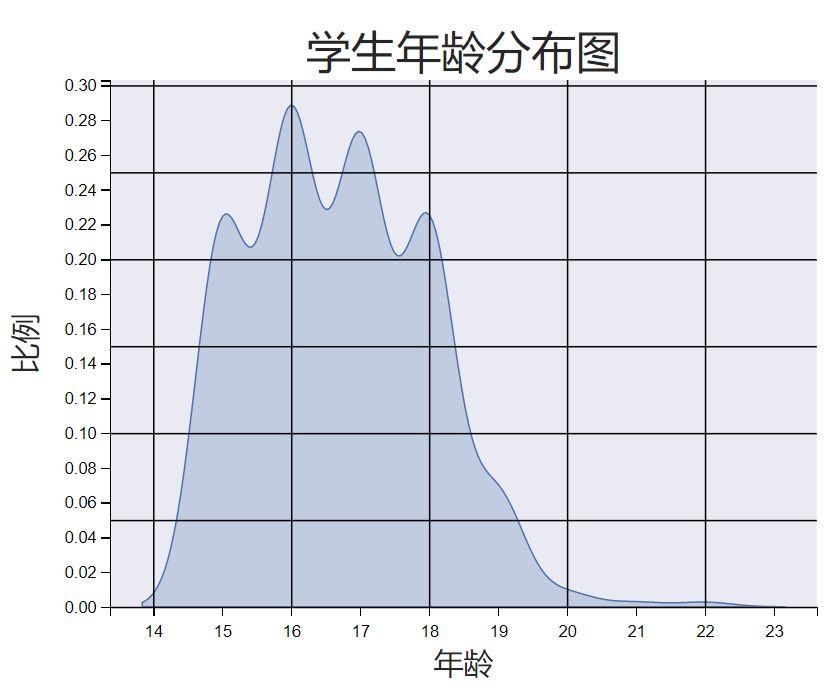
3.3.4分析性别比例



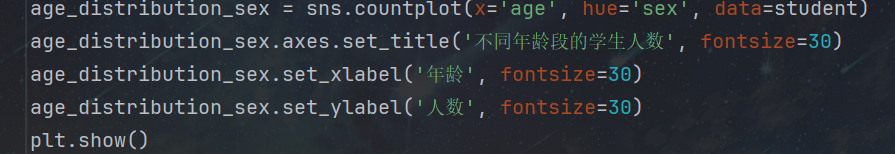


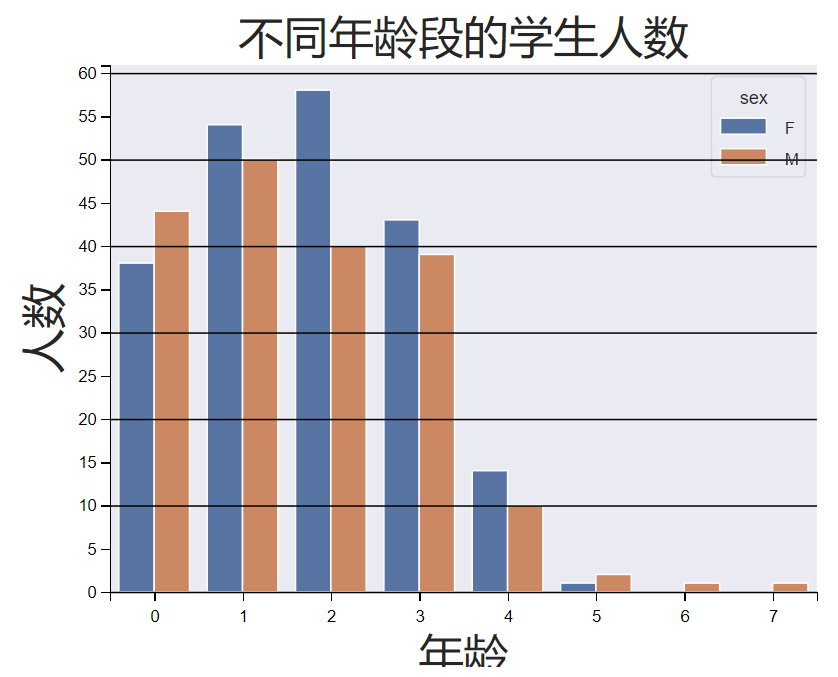
3.3.5 分析年龄分布比例（曲线图）



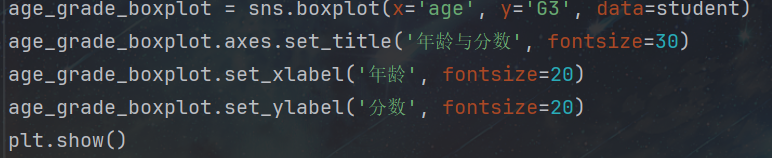


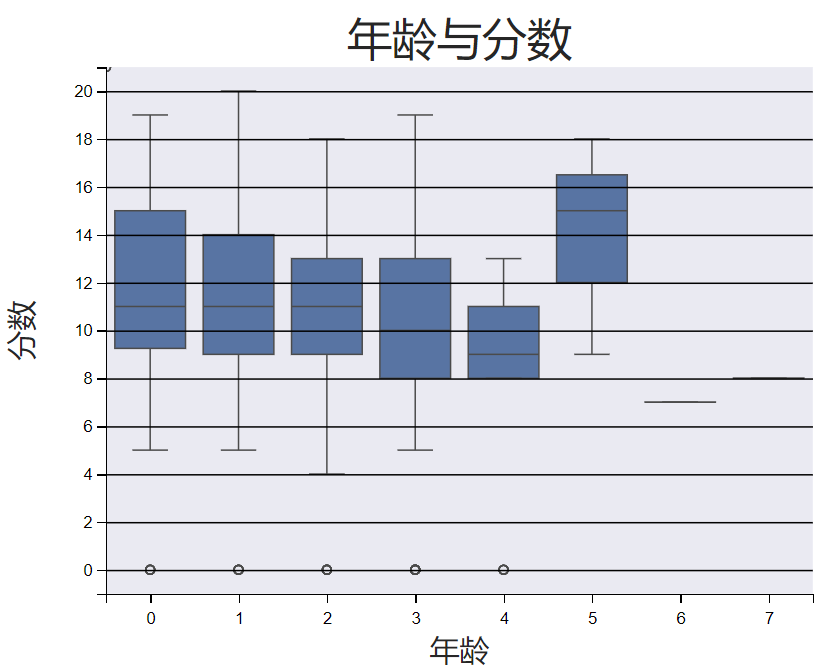
3.3.6 分性别年龄分布图（柱状图）



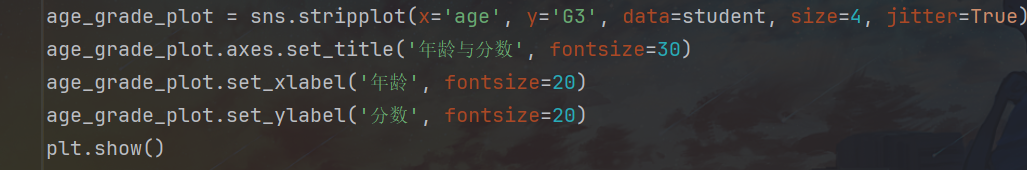


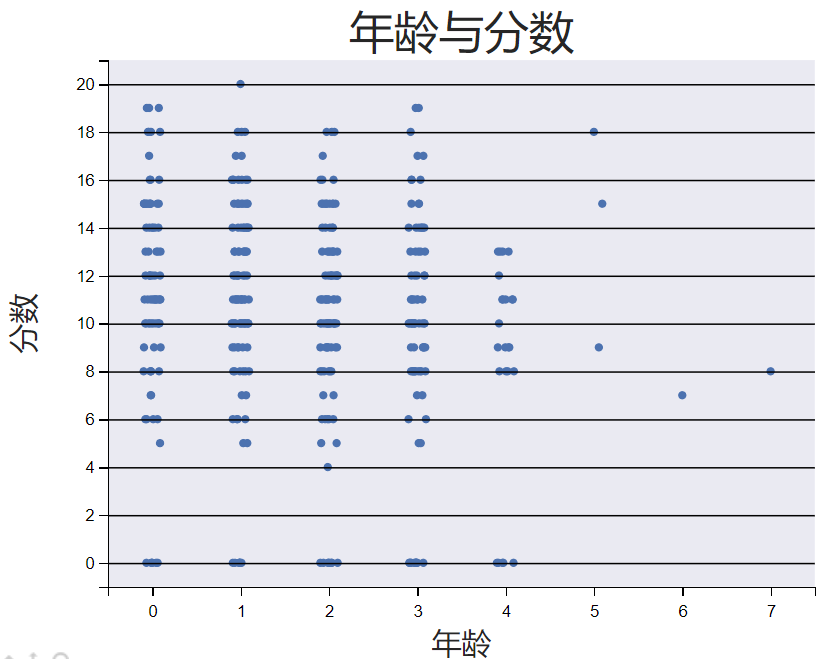
3.3.7 各年龄段的成绩箱型图



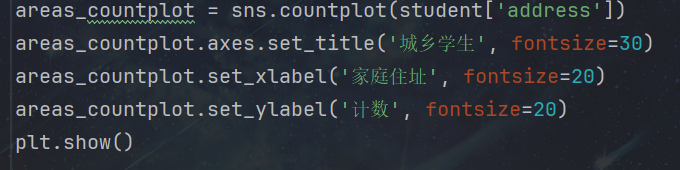


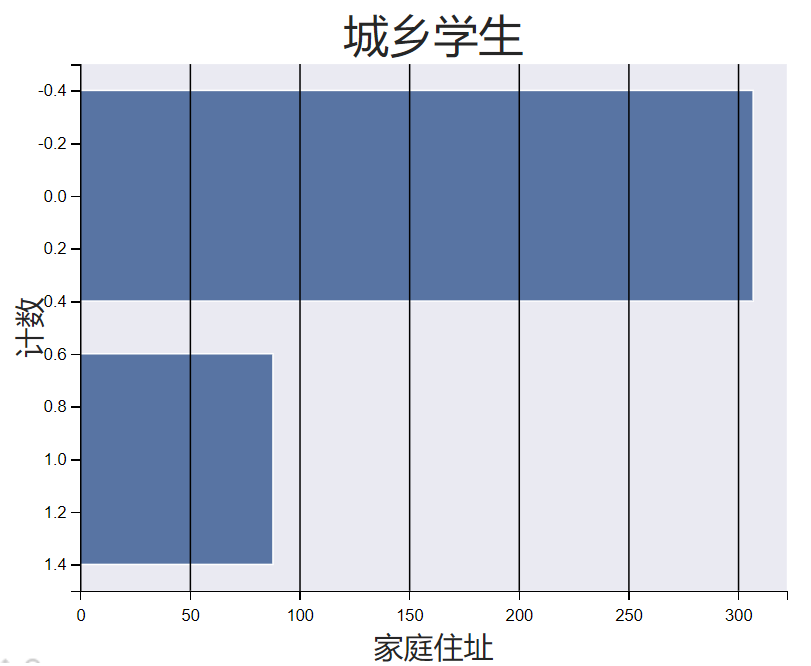
3.3.8 各年龄段的成绩分布图





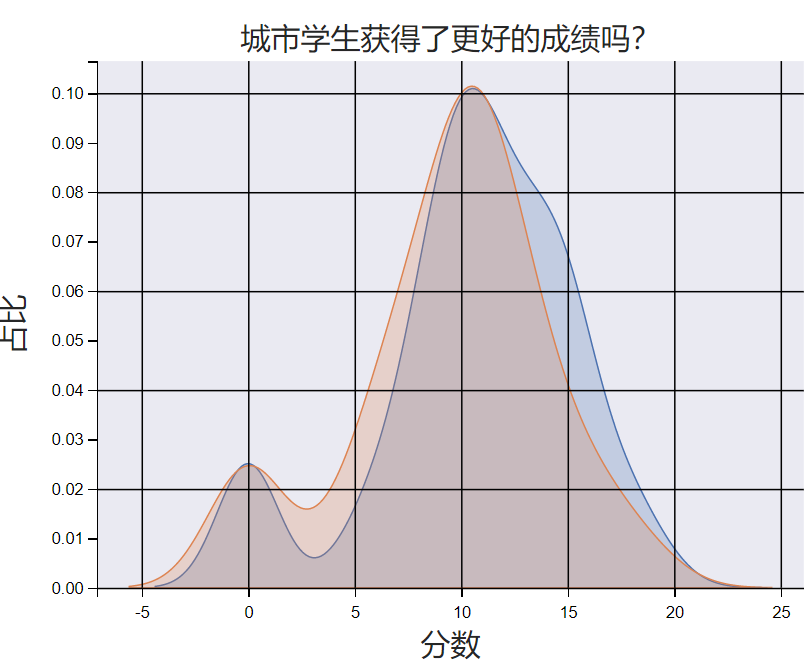
3.3.9 城乡学生计数



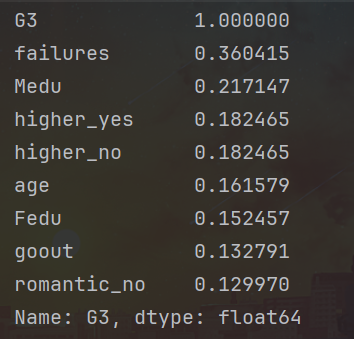


3.3.10 城市学生获得了更好的成绩吗

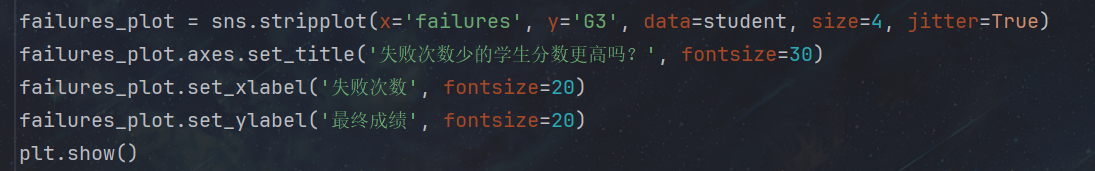


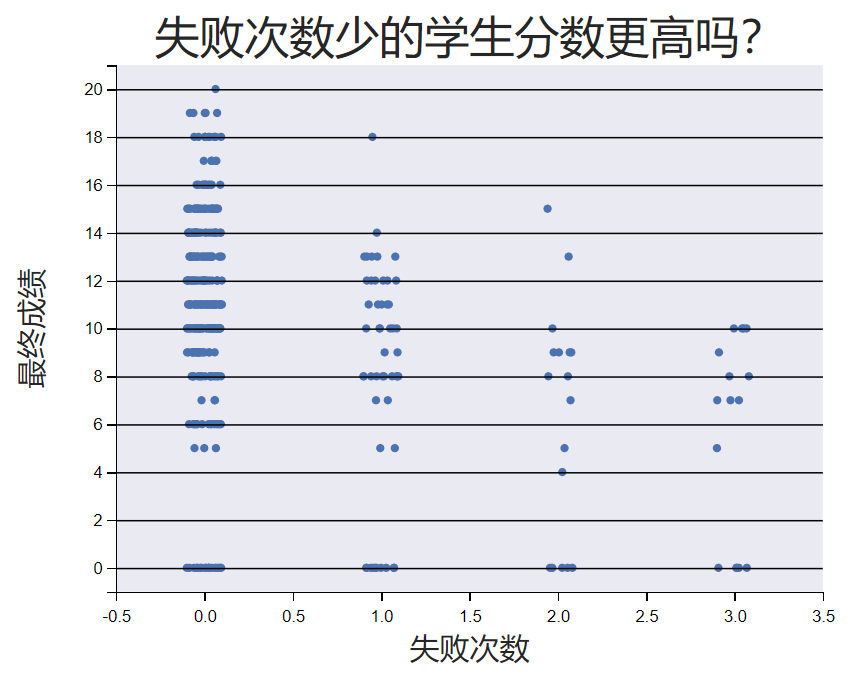


3.3.11 选取G3属性值，选取相关性最强的8个

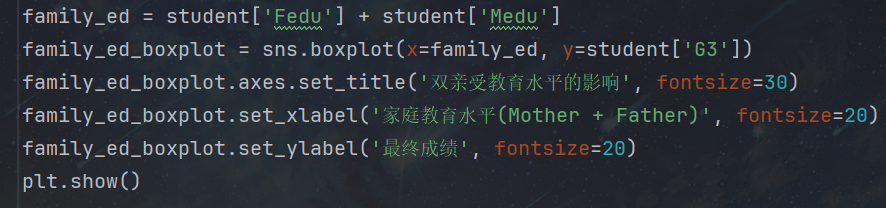
 

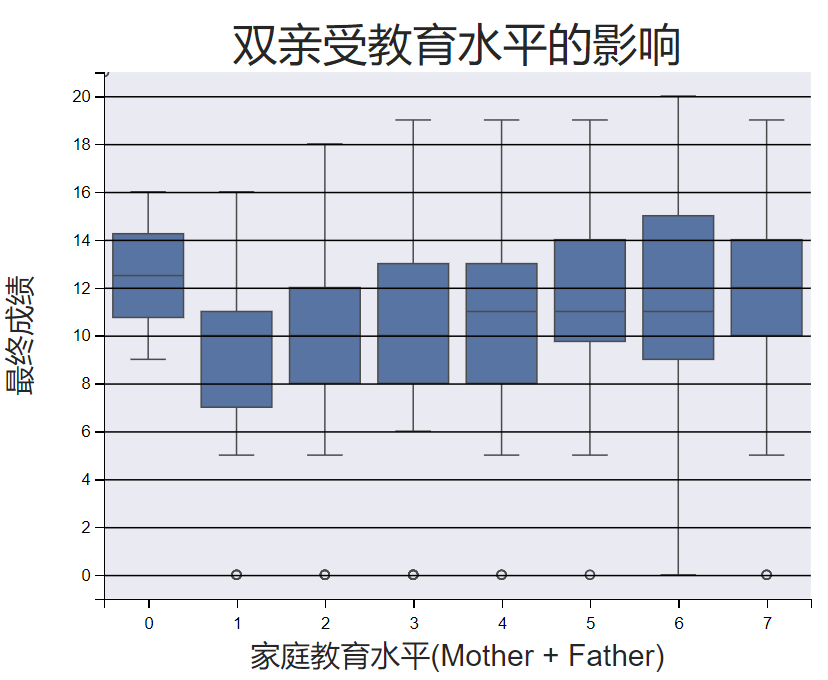
3.3.12 失败次数成绩分布图



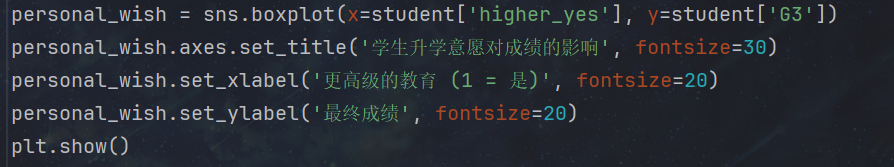


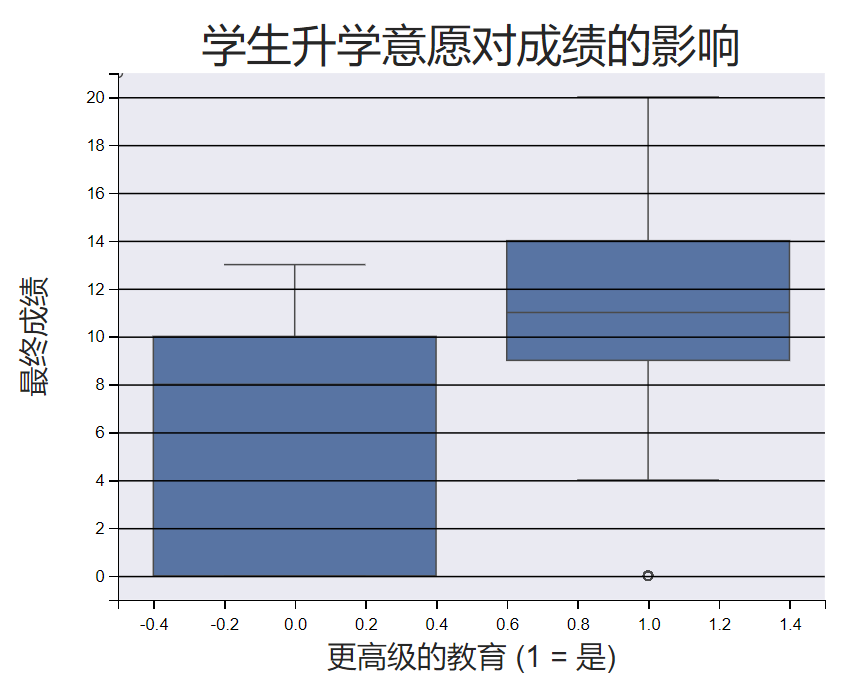
3.3.13 双亲受教育水平的影响





3.3.14 学生自己的升学意志对成绩的影响

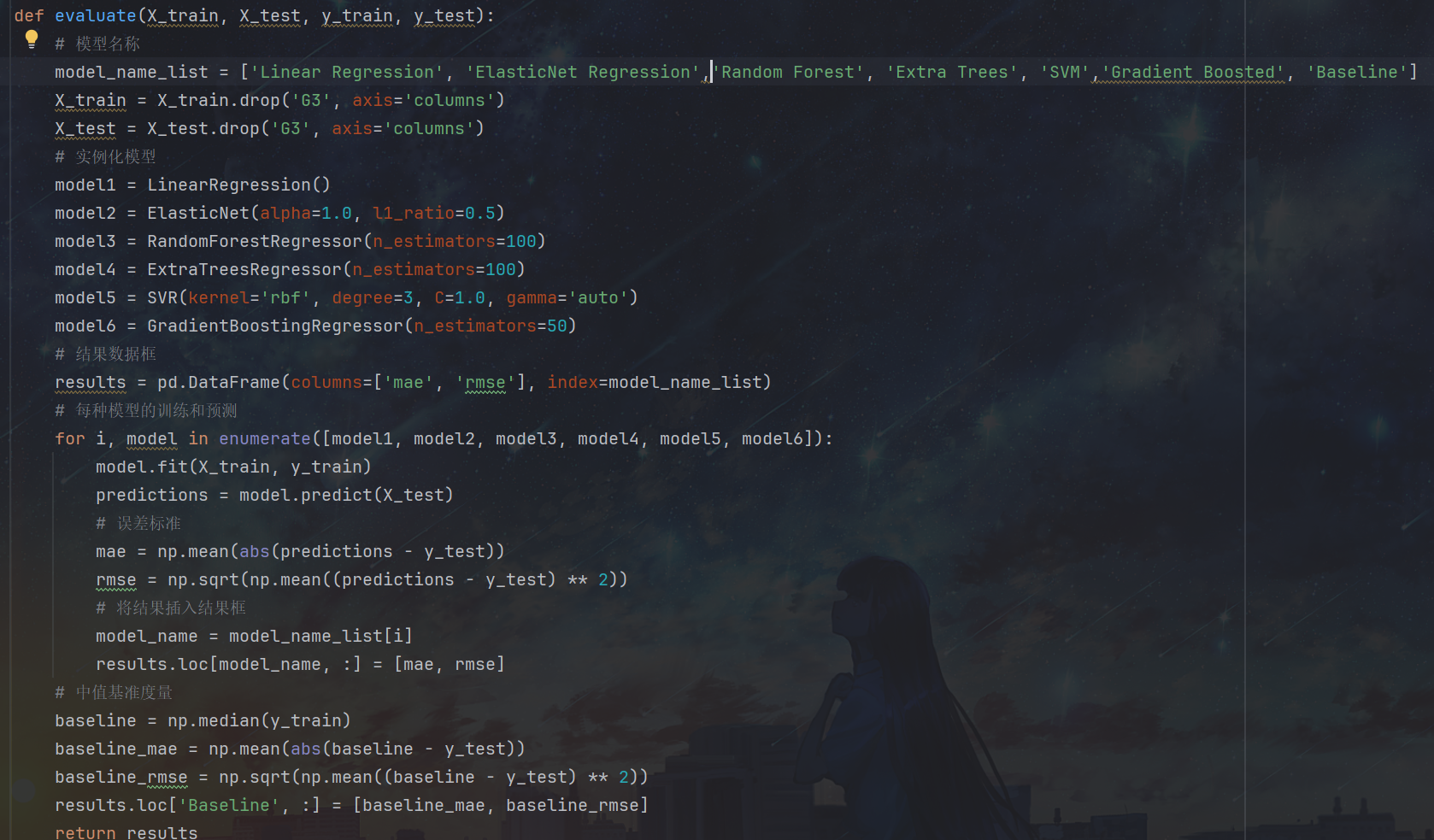


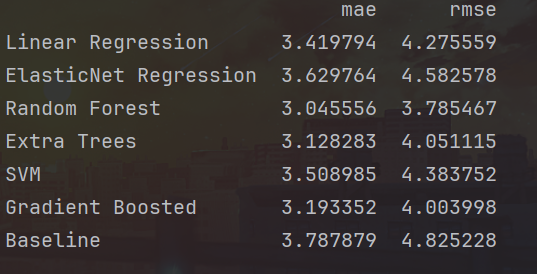


# 分割数据集  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(student, labels, test\_size=0.25, random\_state=42)  
  
  
# 计算平均绝对误差和均方根误差  
# MAE-平均绝对误差  
# RMSE-均方根误差  
def evaluate\_predictions(predictions, true):  
 mae = np.mean(abs(predictions - true))  
 rmse = np.sqrt(np.mean((predictions - true) \*\* 2))  
  
 return mae, rmse  
  
  
# 求中位数  
median\_pred = X\_train['G3'].median()  
  
# 所有中位数的列表  
median\_preds = [median\_pred for \_ in range(len(X\_test))]  
  
# 存储真实的G3值以传递给函数  
true = X\_test['G3']  
  
# 展示基准  
mb\_mae, mb\_rmse = evaluate\_predictions(median\_preds, true)  
print('Median Baseline MAE: {:.4f}'.format(mb\_mae))  
print('Median Baseline RMSE: {:.4f}'.format(mb\_rmse))

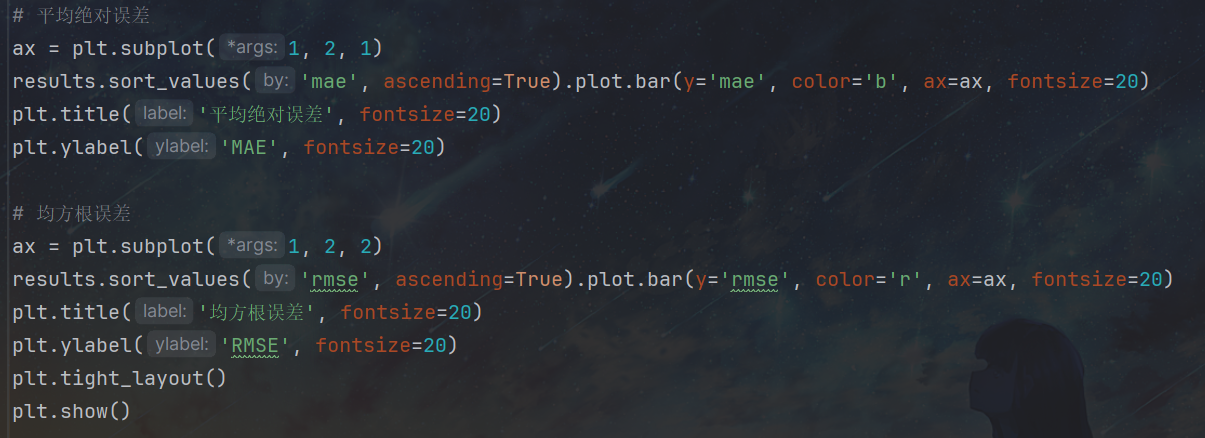


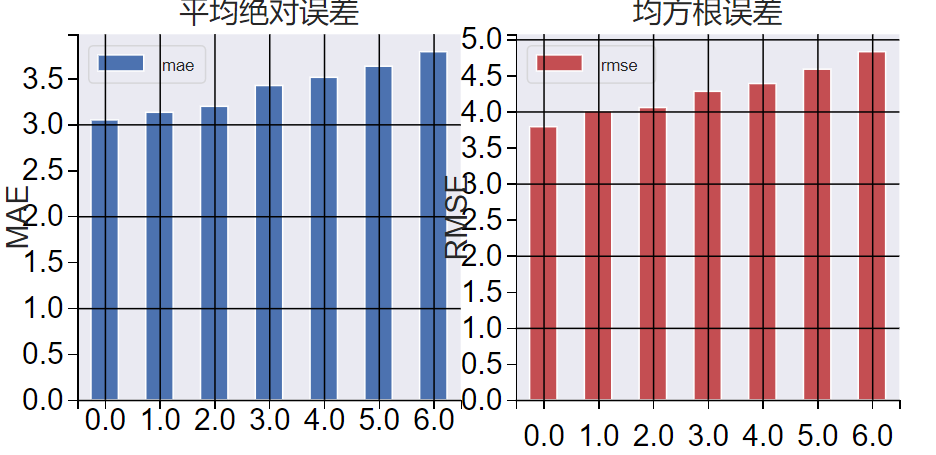
3.3.15 通过训练集训练和测试集测试来生成多个线性模型





3.3.16 得出平均绝对误差和均方根误差





1. **结论**

**1. 学生成绩的多因素影响**：分析结果表明，学生的成绩受到年龄、城乡背景、失败次数、双亲受教育水平和学生升学意愿等多种因素的影响。这强调了教 育过程中需要考虑的多维度因素。

**2. 数据驱动的决策支持**：通过构建的线性回归模型，学校能够基于实证数据做出更加科学的决策，如调整教学方法、优化课程设置等。

**3. 个性化学习路径的指导**：模型预测结果可以帮助学生根据自身情况，如家庭背景、学习习惯等，制定个性化的学习计划，以提高学习效率。

**4. 教育公平性的促进**：研究揭示了不同社会经济背景下学生成绩的差异，为促进教育公平提供了依据，有助于学校和政策制定者采取措施，减少成绩差距。

**5. 持续的模型优化**：虽然当前模型已经能够提供有价值的见解，但随着数据的积累和算法的改进，模型的准确性和预测能力有待进一步提升。

**6. 教育实践的启示**：本研究为教育工作者提供了实证研究基础，有助于他们在教学实践中更好地理解学生需求，实现教学内容和方法的创新。

预测学生成绩的目的是为了更好地理解影响学业表现的因素，并据此采取行动来提高学生的成绩。根据预测结果，为学生提供定制化的学习计划和辅导；对可能落后的学生进行早期识别和干预；根据学生需求调整教学策略和课程内容；为可能面临压力的学生提供心理健康支持等等方式。

**5. 附录**

**5.1实验代码**

import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
from matplotlib.font\_manager import FontProperties  
from sklearn.linear\_model import LinearRegression  
from sklearn.linear\_model import ElasticNet  
from sklearn.ensemble import RandomForestRegressor  
from sklearn.ensemble import ExtraTreesRegressor  
from sklearn.ensemble import GradientBoostingRegressor  
from sklearn.svm import SVR  
from sklearn.model\_selection import train\_test\_split  
from sklearn.preprocessing import MinMaxScaler  
from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, median\_absolute\_error  
import scipy  
import pickle  
  
# 初始化数据  
plt.rcParams['font.sans-serif'] = ['SimHei'] # 中文字体设置-黑体  
plt.rcParams['axes.unicode\_minus'] = False # 解决保存图像是负号'-'显示为方块的问题  
sns.set(font='SimHei') # 解决Seaborn中文显示问题  
student = pd.read\_csv('student-mat.csv')  
#print(student.head())  
  
  
# 分析G3数据属性  
# print(student['G3'].describe())  
  
# 根据人数多少统计各分数段的学生人数  
grade\_counts = student['G3'].value\_counts().sort\_values().plot.barh(width=.9, color=sns.color\_palette('inferno', 40))  
grade\_counts.axes.set\_title('各分数值的学生分布', fontsize=30)  
grade\_counts.set\_xlabel('学生数量', fontsize=30)  
grade\_counts.set\_ylabel('最终成绩', fontsize=30)  
plt.show()  
  
# 从低到高展示成绩分布图  
grade\_distribution = sns.countplot(student['G3'])  
grade\_distribution.set\_title('成绩分布图', fontsize=30)  
grade\_distribution.set\_xlabel('期末成绩', fontsize=20)  
grade\_distribution.set\_ylabel('人数统计', fontsize=20)  
plt.show()  
  
# 检查各个列是否有null值，如果没有表示成绩中的0分确实是0分  
# print(student.isnull().any())  
  
# 分析性别比例  
male\_studs = len(student[student['sex'] == 'M'])  
female\_studs = len(student[student['sex'] == 'F'])  
print('男同学数量:', male\_studs)  
print('女同学数量:', female\_studs)  
  
# 分析年龄分布比例（曲线图）  
age\_distribution = sns.kdeplot(student['age'], fill=True)  
age\_distribution.axes.set\_title('学生年龄分布图', fontsize=30)  
age\_distribution.set\_xlabel('年龄', fontsize=20)  
age\_distribution.set\_ylabel('比例', fontsize=20)  
plt.show()  
  
# 分性别年龄分布图（柱状图）  
age\_distribution\_sex = sns.countplot(x='age', hue='sex', data=student)  
age\_distribution\_sex.axes.set\_title('不同年龄段的学生人数', fontsize=30)  
age\_distribution\_sex.set\_xlabel('年龄', fontsize=30)  
age\_distribution\_sex.set\_ylabel('人数', fontsize=30)  
plt.show()  
  
# 各年龄段的成绩箱型图  
age\_grade\_boxplot = sns.boxplot(x='age', y='G3', data=student)  
age\_grade\_boxplot.axes.set\_title('年龄与分数', fontsize=30)  
age\_grade\_boxplot.set\_xlabel('年龄', fontsize=20)  
age\_grade\_boxplot.set\_ylabel('分数', fontsize=20)  
plt.show()  
  
# 各年龄段的成绩分布图##########################################  
age\_grade\_plot = sns.stripplot(x='age', y='G3', data=student, size=4, jitter=True)  
age\_grade\_plot.axes.set\_title('年龄与分数', fontsize=30)  
age\_grade\_plot.set\_xlabel('年龄', fontsize=20)  
age\_grade\_plot.set\_ylabel('分数', fontsize=20)  
plt.show()  
  
# 城乡学生计数  
areas\_countplot = sns.countplot(student['address'])  
areas\_countplot.axes.set\_title('城乡学生', fontsize=30)  
areas\_countplot.set\_xlabel('家庭住址', fontsize=20)  
areas\_countplot.set\_ylabel('计数', fontsize=20)  
plt.show()  
  
# Grade distribution by address  
sns.kdeplot(student.loc[student['address'] == 'U', 'G3'], label='Urban', fill=True)  
sns.kdeplot(student.loc[student['address'] == 'R', 'G3'], label='Rural', fill=True)  
plt.title('城市学生获得了更好的成绩吗？', fontsize=20)  
plt.xlabel('分数', fontsize=20)  
plt.ylabel('占比', fontsize=20)  
plt.show()  
  
# 选取G3属性值  
labels = student['G3']  
# 删除school，G1和G2属性  
student = student.drop(['school', 'G1', 'G2'], axis='columns')  
# 对离散变量进行独热编码  
student = pd.get\_dummies(student)  
# 选取相关性最强的8个  
most\_correlated = student.corr().abs()['G3'].sort\_values(ascending=False)  
most\_correlated = most\_correlated[:9]  
print(most\_correlated)  
  
# 失败次数成绩分布图################################################  
failures\_plot = sns.stripplot(x='failures', y='G3', data=student, size=4, jitter=True)  
failures\_plot.axes.set\_title('失败次数少的学生分数更高吗？', fontsize=30)  
failures\_plot.set\_xlabel('失败次数', fontsize=20)  
failures\_plot.set\_ylabel('最终成绩', fontsize=20)  
plt.show()  
  
# 双亲受教育水平的影响  
family\_ed = student['Fedu'] + student['Medu']  
family\_ed\_boxplot = sns.boxplot(x=family\_ed, y=student['G3'])  
family\_ed\_boxplot.axes.set\_title('双亲受教育水平的影响', fontsize=30)  
family\_ed\_boxplot.set\_xlabel('家庭教育水平(Mother + Father)', fontsize=20)  
family\_ed\_boxplot.set\_ylabel('最终成绩', fontsize=20)  
plt.show()  
  
# 学生自己的升学意志对成绩的影响  
personal\_wish = sns.boxplot(x=student['higher\_yes'], y=student['G3'])  
personal\_wish.axes.set\_title('学生升学意愿对成绩的影响', fontsize=30)  
personal\_wish.set\_xlabel('更高级的教育 (1 = 是)', fontsize=20)  
personal\_wish.set\_ylabel('最终成绩', fontsize=20)  
plt.show()  
  
# 分割数据集  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(student, labels, test\_size=0.25, random\_state=42)  
  
  
# 计算平均绝对误差和均方根误差  
# MAE-平均绝对误差  
# RMSE-均方根误差  
def evaluate\_predictions(predictions, true):  
 mae = np.mean(abs(predictions - true))  
 rmse = np.sqrt(np.mean((predictions - true) \*\* 2))  
  
 return mae, rmse  
  
  
# 求中位数  
median\_pred = X\_train['G3'].median()  
  
# 所有中位数的列表  
median\_preds = [median\_pred for \_ in range(len(X\_test))]  
  
# 存储真实的G3值以传递给函数  
true = X\_test['G3']  
  
# 展示基准  
mb\_mae, mb\_rmse = evaluate\_predictions(median\_preds, true)  
print('Median Baseline MAE: {:.4f}'.format(mb\_mae))  
print('Median Baseline RMSE: {:.4f}'.format(mb\_rmse))  
  
  
# 通过训练集训练和测试集测试来生成多个线性模型  
def evaluate(X\_train, X\_test, y\_train, y\_test):  
 # 模型名称  
 model\_name\_list = ['Linear Regression', 'ElasticNet Regression','Random Forest', 'Extra Trees', 'SVM','Gradient Boosted', 'Baseline']  
 X\_train = X\_train.drop('G3', axis='columns')  
 X\_test = X\_test.drop('G3', axis='columns')  
 # 实例化模型  
 model1 = LinearRegression()  
 model2 = ElasticNet(alpha=1.0, l1\_ratio=0.5)  
 model3 = RandomForestRegressor(n\_estimators=100)  
 model4 = ExtraTreesRegressor(n\_estimators=100)  
 model5 = SVR(kernel='rbf', degree=3, C=1.0, gamma='auto')  
 model6 = GradientBoostingRegressor(n\_estimators=50)  
 # 结果数据框  
 results = pd.DataFrame(columns=['mae', 'rmse'], index=model\_name\_list)  
 # 每种模型的训练和预测  
 for i, model in enumerate([model1, model2, model3, model4, model5, model6]):  
 model.fit(X\_train, y\_train)  
 predictions = model.predict(X\_test)  
 # 误差标准  
 mae = np.mean(abs(predictions - y\_test))  
 rmse = np.sqrt(np.mean((predictions - y\_test) \*\* 2))  
 # 将结果插入结果框  
 model\_name = model\_name\_list[i]  
 results.loc[model\_name, :] = [mae, rmse]  
 # 中值基准度量  
 baseline = np.median(y\_train)  
 baseline\_mae = np.mean(abs(baseline - y\_test))  
 baseline\_rmse = np.sqrt(np.mean((baseline - y\_test) \*\* 2))  
 results.loc['Baseline', :] = [baseline\_mae, baseline\_rmse]  
 return results  
results = evaluate(X\_train, X\_test, y\_train, y\_test)  
print(results)  
  
# 找出最合适的模型  
plt.figure(figsize=(12, 8))  
  
# 平均绝对误差  
ax = plt.subplot(1, 2, 1)  
results.sort\_values('mae', ascending=True).plot.bar(y='mae', color='b', ax=ax, fontsize=20)  
plt.title('平均绝对误差', fontsize=20)  
plt.ylabel('MAE', fontsize=20)  
  
# 均方根误差  
ax = plt.subplot(1, 2, 2)  
results.sort\_values('rmse', ascending=True).plot.bar(y='rmse', color='r', ax=ax, fontsize=20)  
plt.title('均方根误差', fontsize=20)  
plt.ylabel('RMSE', fontsize=20)  
plt.tight\_layout()  
plt.show()  
  
# 保存线性回归模型  
model = LinearRegression()  
model.fit(X\_train, y\_train)  
filename = 'LR\_Model'  
pickle.dump(model, open(filename, 'wb'))

**5.2数据集的属性介绍**

1.学校-学生学校（二进制：“ GP”-加布里埃尔·佩雷拉（Gabriel Pereira）或“ MS”-Mousinho da Silveira）

2.性别-学生的性别（二进制：“ F”-女性或“ M”-男性）

3.年龄-学生的年龄（数字：15至22）

4.地址-学生的家庭住址类型（二进制：“ U”-城市或“ R”-农村）

5.famsize-家庭大小（二进制：“ LE3”-小于或等于3或“ GT3”-大于3）

6.Pstatus-父母的同居状态（二进制：“ T”-同居或“ A”-分开）

7.Medu-母亲的教育（数字：0-无，1-初等教育（四年级），2 – 5至9年级，3 –中等教育或4 –高等教育）

8.Fedu-父亲的教育（数字：0-无，1-初等教育（四年级），2 – 5至9年级，3 –中等教育或4 –高等教育）

9.Mjob-母亲的工作（名义：“教师”，“与健康”有关的，民事“服务”（例如行政或警察），“在家”或“其他”）

10.Fjob-父亲的工作（名义：“教师”，“与健康”相关的，民事“服务”（例如行政或警察），“在家”或“其他”）

11.理由-选择这所学校的理由（名义：接近“家”，学校“声誉”，“课程”偏好或“其他”）

12.监护人-学生的监护人（名词：“母亲”，“父亲”或“其他”）

13.traveltime-学校到学校的旅行时间（数字：1-<15分钟，2-15至30分钟，3-30分钟至1小时或4-> 1小时）

14.学习时间-每周学习时间（数字：1-<2小时，2-2至5小时，3-5至10小时或4-> 10小时）

15.失败-过去类失败的次数（数字：如果1 <= n ，则为n，否则为4）

16.schoolup-额外的教育支持（二进制：是或否）

17.famsup-家庭教育支持（二进制：是或否）

18.付费-课程主题内的额外付费课程（数学或葡萄牙语）（二进制：是或否）

19.活动-课外活动（二进制：是或否）

20.托儿所-上托儿所（二进制：是或否）

21.更高-想要接受高等教育（二进制：是或否）

22.互联网-在家上网（二进制：是或否）

23.浪漫-具有浪漫关系（二进制：是或否）

24.家族-家庭关系的质量（数字：从1-非常差到5-极好）

25.空闲时间-放学后的空闲时间（数字：从1-非常低到5-非常高）

26.外出-与朋友外出（数字：从1-非常低到5-非常高）

27.Dalc-工作日酒精消耗（数字：从1-非常低到5-非常高）

28.Walc-周末酒精消耗（数字：从1-非常低至5-非常高）

29.健康-当前的健康状况（数字：从1-非常差到5-非常好）

30.缺勤-缺勤人数（数字：0到93）

这些成绩与课程主题（数学或葡萄牙语）相关：

31.G1-第一期成绩（数字：0至20）

32.G2-第二学期成绩（数字：0至20）

33.G3-最终成绩（数字：0到20，输出目标）