一、具体实现步骤

1.导入数据 ¶

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt

cars = pd.read_csv('../data/auto-mpg.data',names=["燃油效率","气缸","排量","马力","重量","加速度","型号年份","编号","原产地"],delim_whitespace = True)
cars.head()
```

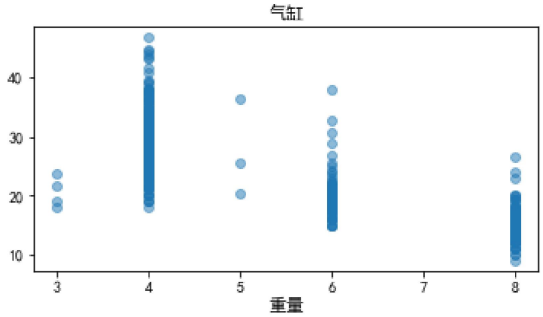
Out[1]:

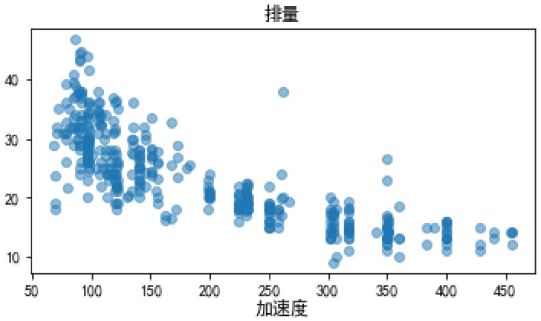
	燃油效率	气缸	排量	马力	重量	加速度	型号年份	编号	原产地
0	18.0	8	307.0	130.0	3504.0	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693.0	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150.0	3436.0	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150.0	3433.0	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140.0	3449.0	10.5	70	1	ford torino

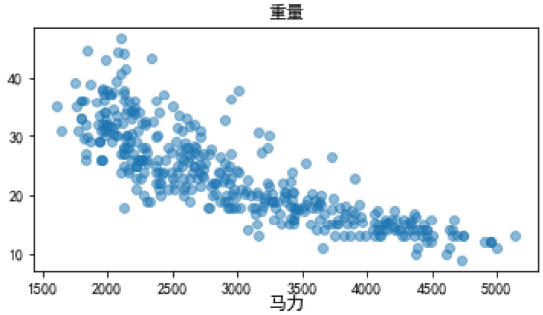
2.探究数据关系

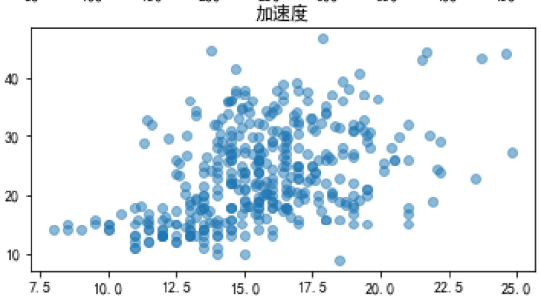
In [4]:

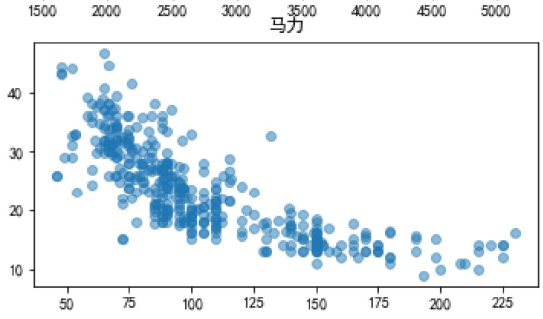
```
import numpy as np
import matplotlib.ticker as ticker
#删除horsepower值为'?'的行
cars = cars[cars.马力 != '?']
#设置中文显示
from pylab import mpl
mpl.rcParams['font.sans-serif'] = ['SimHei']

#用散点图分别展示气缸、排量、重量、加速度与燃油效率的关系
fig = plt.figure(figsize = (13,10))
ax1 = fig.add_subplot(321)
ax2 = fig.add_subplot(322)
ax3 = fig.add_subplot(323)
ax4 = fig.add_subplot(324)
ax5 = fig.add_subplot(325)
ax1.scatter(cars['气缸'], cars['燃油效率'], alpha=0.5)
ax1.set_title('气缸')
ax2.scatter(cars['排量'], cars['燃油效率'], alpha=0.5)
ax2.set_title('排量')
ax3.scatter(cars['重量'], cars['燃油效率'], alpha=0.5)
ax3.set_title('重量')
ax4.scatter(cars['加速度'], cars['燃油效率'], alpha=0.5)
ax4.set_title('加速度')
ax5.scatter([ float(x) for x in cars['马力'].tolist()], cars['燃油效率'], alpha=0.5)
ax5.set_title('马力')
plt.show()
```

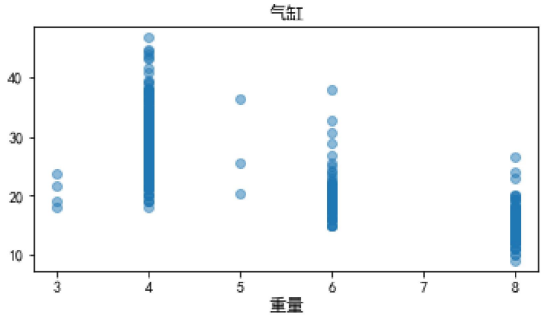

气缸

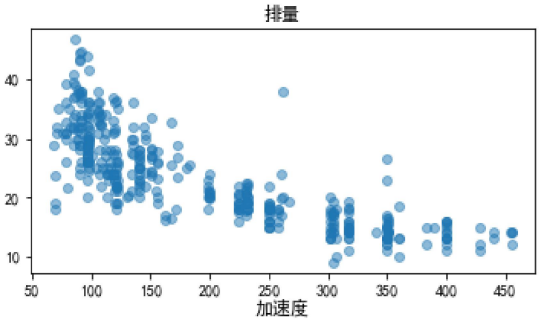

排量

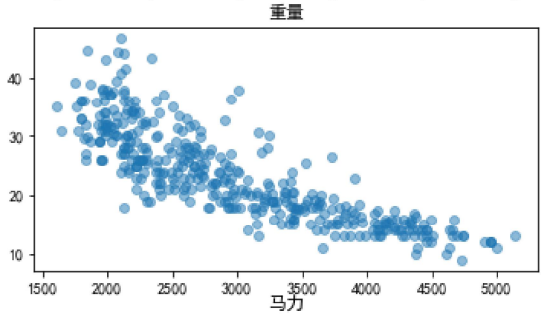

重量

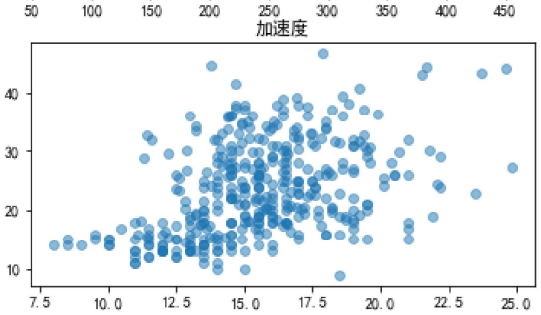

加速度

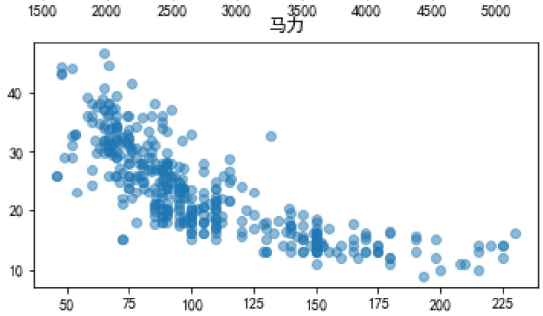

马力

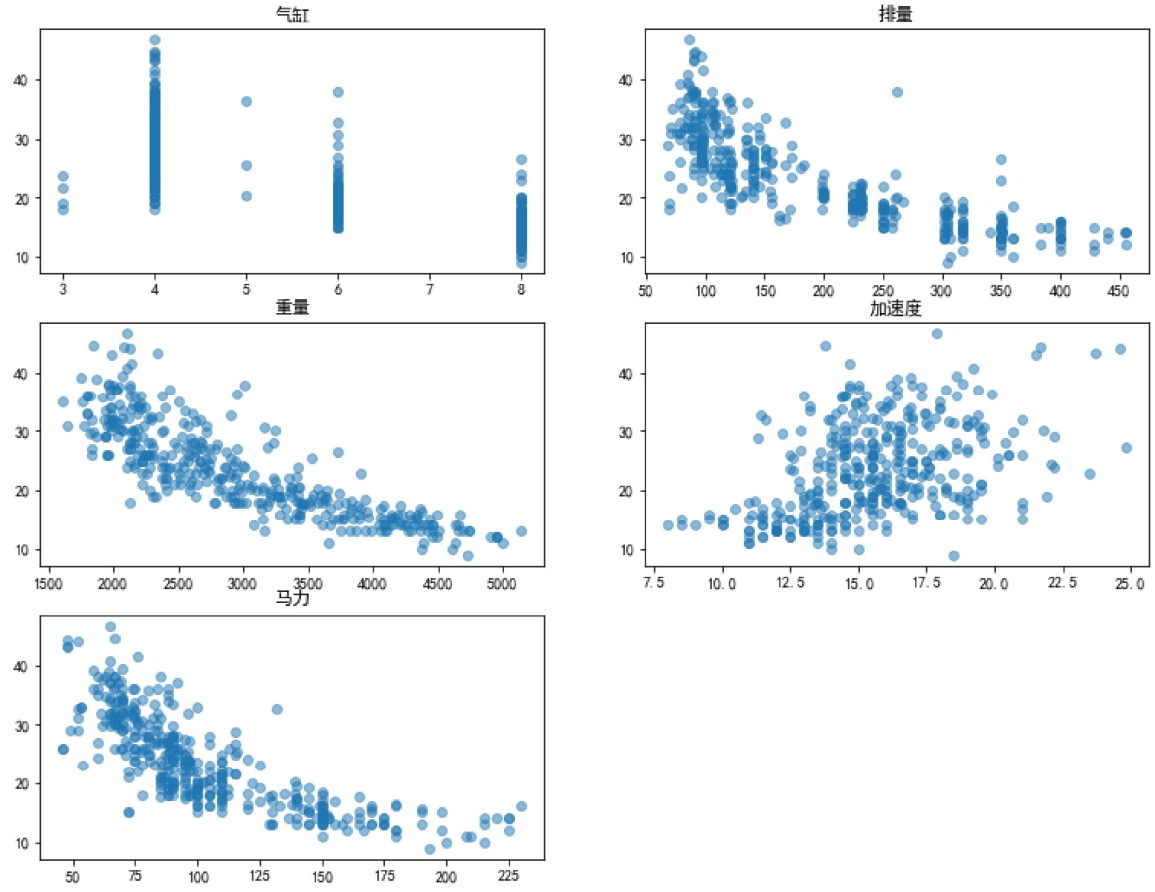

气缸


排量


重量


加速度


马力



3.提取数据

从上面我们已经可以看出汽车与燃油有着线性关系，下面我们将用这部门数据进行训练

In [6]:

```
Y = cars['燃油效率']
X = cars['重量']
X = X.to_numpy(X)
Y = Y.to_numpy(Y)
X = X.reshape(len(X),1)
Y = Y.reshape(len(Y),1)
```

4.拆分数据

In [7]:

```
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)
```

5.训练模型

In [8]:

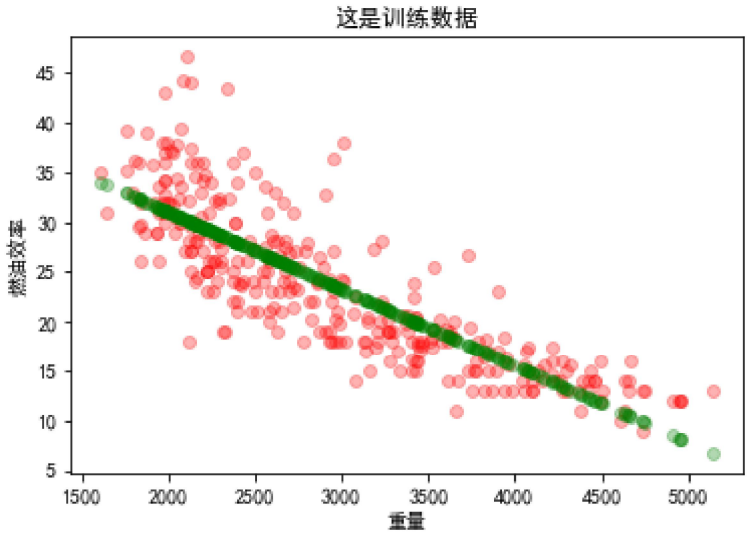
```
from sklearn.linear_model import LinearRegression
LR = LinearRegression()
LR = LR.fit(X_train, Y_train)
```

二、可视化结果展示

1.训练集

In [9]:

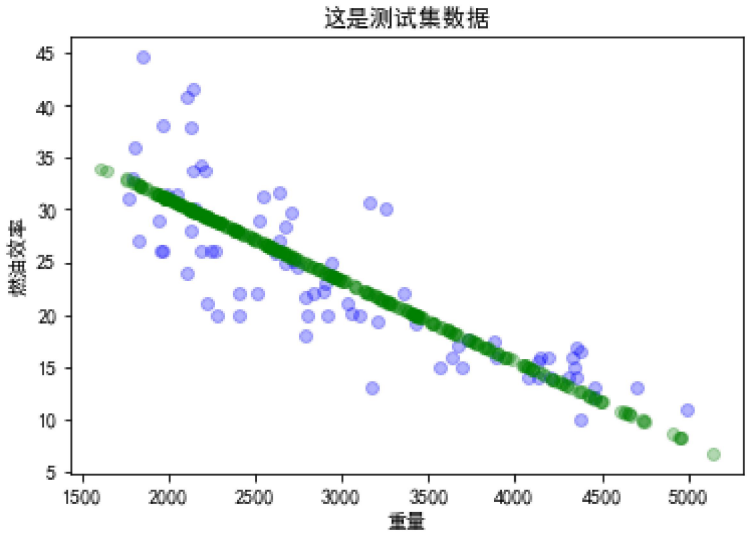
```
import matplotlib.pyplot as plt

plt.scatter(X_train, Y_train, color='red', alpha=0.3)
plt.scatter(X_train, LR.predict(X_train), color='green', alpha=0.3)
plt.xlabel("重量")
plt.ylabel("燃油效率")
plt.title("这是训练数据")
plt.show()
```



2. 测试集

In [10]:

```
plt.scatter(X_test, Y_test, color='blue', alpha=0.3)
plt.scatter(X_train, LR.predict(X_train), color='green', alpha=0.3)
plt.xlabel("重量")
plt.ylabel("燃油效率")
plt.title("这是测试集数据")
plt.show()
```



3.计算模型得分

In [11]:

```
score = LR.score(cars[['重量']],cars['燃油效率'])
score
```

Out[11]:

0.692564100650704

三、多元线性回归

1.训练模型

In [12]:

```
#初始化模型
mul_LR_model = LinearRegression()
#拟合模型
mul_LR_model.fit(cars[['重量','马力','排量']], cars['燃油效率'])
#预测
cars['燃料效率预测值'] = mul_LR_model.predict(cars[['重量','马力','排量']])
#显示
cars.head(5)
```

Out[12]:

	燃油效率	气缸	排量	马力	重量	加速度	型号年份	编号	原产地	燃料效率预测值
0	18.0	8	307.0	130.0	3504.0	12.0	70	1	chevrolet chevelle malibu	18.915289
1	15.0	8	350.0	165.0	3693.0	11.5	70	1	buick skylark 320	16.197184
2	18.0	8	318.0	150.0	3436.0	11.0	70	1	plymouth satellite	18.382258
3	16.0	8	304.0	150.0	3433.0	12.0	70	1	amc rebel sst	18.479076
4	17.0	8	302.0	140.0	3449.0	10.5	70	1	ford torino	18.821729

2.计算得分

In [13]:

```
mul_score = mul_LR_model.score(cars[['重量','马力','排量']], cars['燃油效率'])
mul_score
```

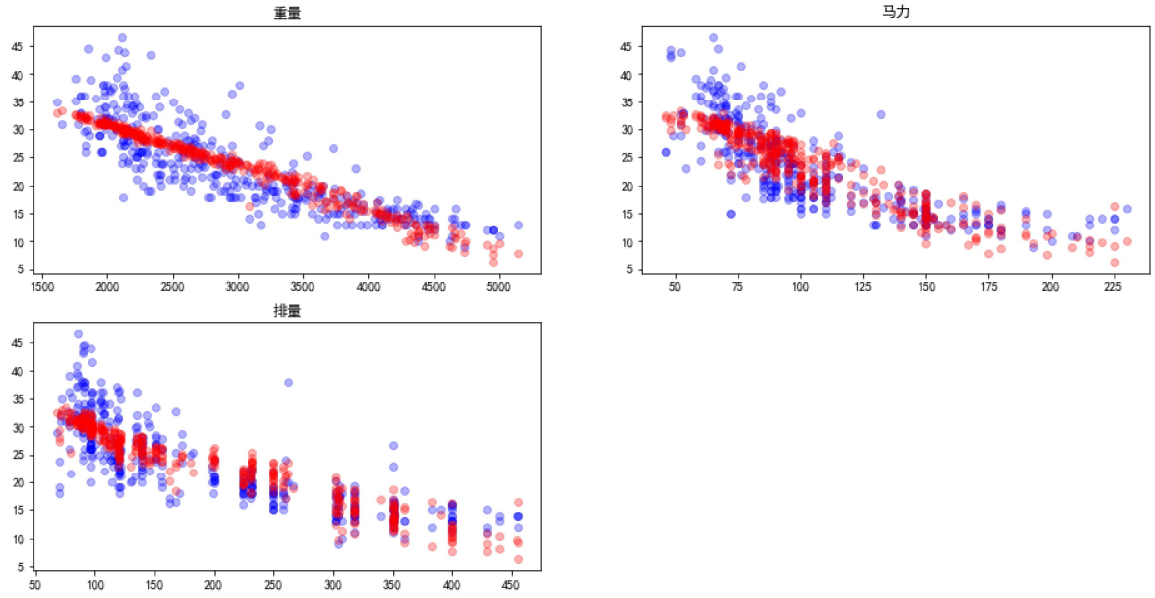
Out[13]:

0.7069554693444708

3.可视化预测结果

In [14]:

```
fig = plt.figure(figsize = (16,8))
ax1 = fig.add_subplot(2,2,1)
ax2 = fig.add_subplot(2,2,2)
ax3 = fig.add_subplot(2,2,3)
ax1.scatter(cars['重量'], cars['燃油效率'], c='blue', alpha=0.3)
ax1.scatter(cars['重量'], cars['燃料效率预测值'], c='red', alpha=0.3)
ax1.set_title('重量')
ax2.scatter([ float(x) for x in cars['马力'].tolist()], cars['燃油效率'], c='blue', alpha=0.3)
ax2.scatter([ float(x) for x in cars['马力'].tolist()], cars['燃料效率预测值'], c='red', alpha=
0.3)
ax2.set_title('马力')
ax3.scatter(cars['排量'], cars['燃油效率'], c='blue', alpha=0.3)
ax3.scatter(cars['排量'], cars['燃料效率预测值'], c='red', alpha=0.3)
ax3.set_title('排量')
plt.show()
```



In [ ]: