实验目录

1. 基于多元线性回归的汽车油耗预测模型

实验内容

1. 基于多元线性回归的汽车油耗预测模型

知识点

1) 在 statsmodels 中，直接使用 C(var)标记离散变量

2) 在 statsmodels 中无需使用 one-hot 等方式转换哑变量

3) RMES 可以比较模型准确率

实验目的

1) 学习使用 statsmodels 处理离散变量

2) 学习解释离散变量输出结果

3) 学习使用 one-hot 方法将离散变量转换为哑变量

4) 学习建立多元线性回归模型预测汽车油耗

实验步骤

1) 打开 Jupyter，并新建 python 工程

2) 读取数据

1. Jupyter 输入代码后，使用 shift+enter 执行，下同。

2. 数据来自 1974 年的汽车趋势美国杂志，包括燃料消耗和 32 辆汽车的汽车设计和性能的 10 个方面（1973-1974 车型），本实验保留字段如下:

   mpg: Miles/(US) gallon

   hp: Gross horsepower

   vs: Engine (0 = V-engine, 1 = straight engine)

   am: Transmission (0 = automatic, 1 = manual)

3. 使用 pandas 读取 csv 文件

[Code 001]:

```
import pandas as pd
df = pd.read_csv('/root/experiment/datas/mtcars_p2.csv')
# 查看 df 的维度
df.shape
```

```
import pandas as pd
df = pd.read_csv('/root/experiment/datas/mtcars_p2.csv')
df.shape
```

```
(32, 4)
```

3) 描述性分析与可视化分析

1. 查看 df 的前五项

[Code 002]:

```
df.head()
```

```
df.head()
```

	mpg	hp	vs	am
0	21.0	110	0	1
1	21.0	110	0	1
2	22.8	93	1	1
3	21.4	110	1	0
4	18.7	175	0	0

2. 查看 df 中连续变量的统计描述

[Code 003]:

df.describe()

```
df.describe()
```

	mpg	hp	vs	am
count	32.000000	32.000000	32.000000	32.000000
mean	20.090625	146.687500	0.437500	0.406250
std	6.026948	68.562868	0.504016	0.498991
min	10.400000	52.000000	0.000000	0.000000
25%	15.425000	96.500000	0.000000	0.000000
50%	19.200000	123.000000	0.000000	0.000000
75%	22.800000	180.000000	1.000000	1.000000
max	33.900000	335.000000	1.000000	1.000000

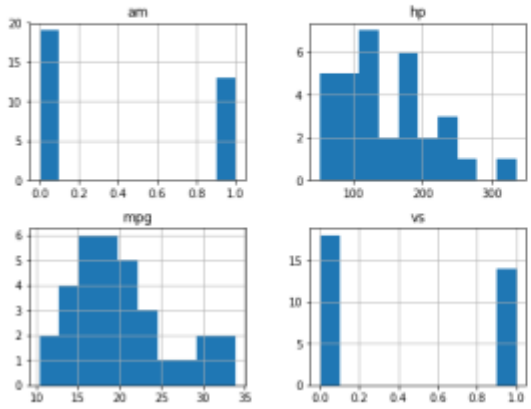
3. 使用 matplotlib 库进行可视化分析（绘图时，由于 jupyter 的问题，执行时可能需重复执行才能显示绘图结果，下同）

[Code 004]:

import matplotlib.pyplot as plt
df.hist(figsize=(8,6))
plt.show()

```
import matplotlib.pyplot as plt
```

```
df.hist(figsize=(8,6))
plt.show()
```



4) 数据处理

1. 为原始数据增加截距项

[Code 005]:

import statsmodels.api as sm

x = df.loc[:,df.columns != 'mpg']

x = sm.add_constant(x)

y = df['mpg']

x.sample(6)

```
import statsmodels.api as sm
```

```
x = df.loc[:,df.columns != 'mpg']
x = sm.add_constant(x)
y = df['mpg']
x.sample(6)
```

	const	hp	vs	am
17	1.0	66	1	1
3	1.0	110	1	0
7	1.0	62	1	0
5	1.0	105	1	0
28	1.0	264	0	1
19	1.0	65	1	1

5) 建立模型

1. 使用 OLS 方法建立线性回归模型

[Code 006]:

import statsmodels.formula.api as smf

model = smf.ols(formula='mpg ~ hp + C(vs) + C(am)',data=df).fit()

 # 查看模型结果

print(model.summary())

```
import statsmodels.formula.api as smf
```

```
model = smf.ols(formula='mpg ~ hp + C(vs) + C(am)',data=df).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.806
Model:                            OLS   Adj. R-squared:                  0.785
Method:                 Least Squares   F-statistic:                     38.68
Date:                Tue, 15 May 2018   Prob (F-statistic):           4.31e-10
Time:                        05:10:38   Log-Likelihood:                -76.171
No. Observations:                  32   AIC:                             160.3
Df Residuals:                      28   BIC:                             166.2
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     23.3342      2.233     10.450      0.000      18.760      27.908
C(vs)[T.1]     2.6588      1.442      1.843      0.076      -0.296       5.614
C(am)[T.1]     5.2985      1.038      5.107      0.000       3.173       7.424
hp            -0.0447      0.011     -4.150      0.000      -0.067      -0.023
==============================================================================
Omnibus:                        0.663   Durbin-Watson:                   1.603
Prob(Omnibus):                  0.718   Jarque-Bera (JB):                0.681
Skew:                           0.040   Prob(JB):                        0.711
Kurtosis:                       2.290   Cond. No.                         835.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

6) 模型预测

1. 使用 predict 对 x 进行预测，并随机查看五项

[Code 007]:

y_hat = model.predict(x)
y_hat.sample(5)

```
y_hat = model.predict(x)
y_hat.sample(5)
```

```
11    15.285231
30    13.652713
18    28.966324
3     21.074231
31    26.417485
dtype: float64
```

2. 计算 RMSE

[Code 008]:

import numpy as np
model_RMSE = np.sqrt(np.mean(np.square(y_hat-y)))
model_RMSE

```
import numpy as np
```

```
model_RMSE = np.sqrt(np.mean(np.square(y_hat-y)))
model_RMSE
```

```
2.615338057939008
```

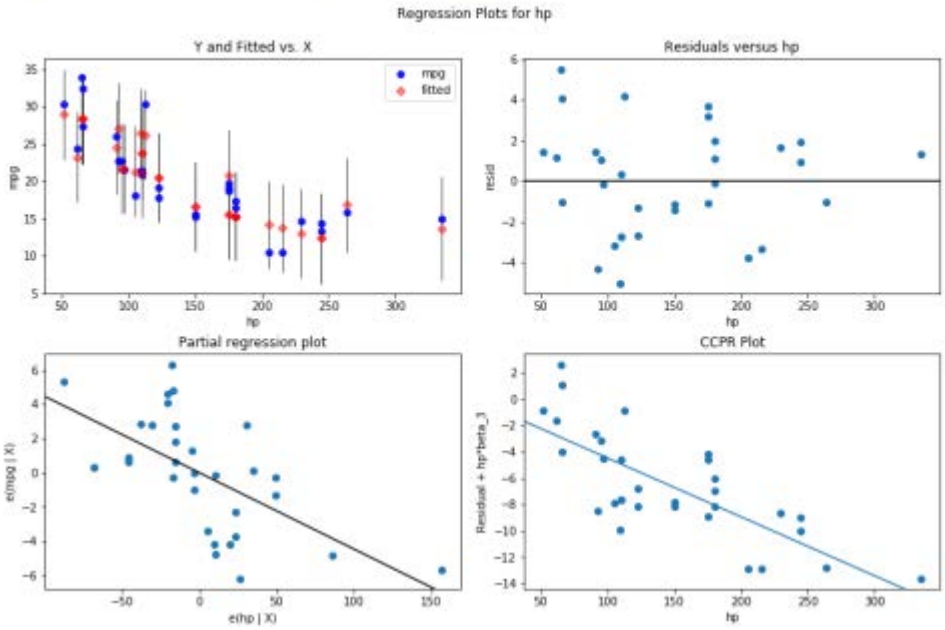
7) 回归诊断

1. 绘制线性回归模型诊断

[Code 009]:

fig = plt.figure(figsize=(12,8))
fig = sm.graphics.plot_regress_exog(model_result, "height", fig=fig)

```
fig = plt.figure(figsize=(12,8))
fig = sm.graphics.plot_regress_exog(model, "hp", fig=fig)
```



8) 实验结论

1. 模型结果显示 Prob 为 4.31e-10，在 0.05 水平显著。
2. R-squared 为 0.806，模型可以解释 80.6%的信息。
3. 截距项、am、hp 在 0.05 水平显著，vs 在 0.1 水平显著。
4. 回归诊断并无明显违背模型假设。
5. 总马力 HP 每增加一个单位，每加仑行驶里程下降 0.0447 个单位。
6. Engine 为 straight engine，比 V-engine 每加仑行驶里程增加 2.6588 个单位。
7. Transmission 为手动挡比自动挡每加仑行驶里程增加 5.2985 个单位。