multiple linear regression 多元线性回归

最开始，进行数据预处理

首先，导入数据集：

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('50_Startups.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

数据集如下图，共50个数据项，特征分别为：

R&D Spend(研发花费) Administration(管理经费) Marketing Spend(市场花费)

要预测的内容为 Profit(盈利)

R&D Spend	Administration	Marketing Spend	State	Profit
165349.2	136897.8	471784.1	New York	192261.8
162597.7	151377.59	443898.53	California	191792.1
153441.51	101145.55	407934.54	Florida	191050.4
144372.41	118671.85	383199.62	New York	182902
142107.34	91391.77	366168.42	Florida	166187.9
131876.9	99814.71	362861.36	New York	156991.1
134615.46	147198.87	127716.82	California	156122.5
130298.13	145530.06	323876.68	Florida	155752.6
120542.52	148718.95	311613.29	New York	152211.8
123334.88	108679.17	304981.62	California	149760
101913.08	110594.11	229160.95	Florida	146122
100671.96	91790.61	249744.55	California	144259.4
93863.75	127320.38	249839.44	Florida	141585.5
91992.39	135495.07	252664.93	California	134307.4
119943.24	156547.42	256512.92	Florida	132602.7
114523.61	122616.84	261776.23	New York	129917
78013.11	121597.55	264346.06	California	126992.9
94657.16	145077.58	282574.31	New York	125370.4
91749.16	114175.79	294919.57	Florida	124266.9
86419.7	153514.11	0	New York	122776.9
76253.86	113867.3	298664.47	California	118474
78389.47	153773.43	299737.29	New York	111313
73994.56	122782.75	303319.26	Florida	110352.3
67532.53	105751.03	304768.73	Florida	108734

然后，建造分类变量的dummy variable：

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelEncoder_X = LabelEncoder()
X[:,3] = labelEncoder_X.fit_transform(X[:,3])
onehotencoder = OneHotEncoder(categorical_features=[3])
X = onehotencoder.fit_transform(X).toarray()
```

为了防止dummy variable trap(虚拟变量陷阱)所产生的Multicollinearity(多重共线性)，需要将其避免，具体方法就是把每组dummy variable的其中一列移除：

```
X = X[:,1:]
```

将数据集分为训练集与测试集：

```
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 0)
```

下面是基础的多元回归模型的训练过程

首先，用多线性回归器对训练集进行拟合，并用在测试集上进行训练：

```
#Fitting Multiple Linear Regression to the  Training set
from sklearn.linear_model import  LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

#Predicting the Test set results
y_pred = regressor.predict(X_test)
```

由于在上一次的简单线性回归中给出了图片示例，在此就不进行展示了。

下面是逐步回归，采用Backward Elimination(反向淘汰)

由于在回归过程中，有很多变量是不需要的(p值较高)，所以要将其淘汰，具体步骤如下：

step 1:为p值选择一个阈值SL(Significance leve)，这里为0.05

step 2:使用所有的可用的变量，训练出模型

step 3:如果p值最高的一个变量，如果其p值 P>SL，跳到step 4，否则跳到最后

step 4:将此变量删除

step 5:利用剩下的变量拟合模型

就如此，不断的循环以上5步，直到没有一个变量的p值大于SL，就停止。

以下为具体过程：

首先导入需要使用的工具库(这里使用statsmodels工具库，因为其可以查看统计数值)，并将所有变量放入数据集中，即X_opt。

其中，需要使用np.append将一列1放入数据集的第一列代表其bias，即截距。即在一列$[1, 1, 1, 1...]^T$的后面加上X

```python
import statsmodels.formula.api as sm
X = np.append(values=X,arr=np.ones((50,1)).astype(int),axis=1)
X_opt = X[:, [0,1,2,3,4,5]]
```

建立模型，并使用所有变量拟合模型，并使用regressor_OLS.summary()查看其p值

```python
regressor_OLS = sm.OLS(endog=y, exog=X_opt).fit()
regressor_OLS.summary()
```

结果显示如下，可看出，x2的p值为0.990，是最大的，且大于0.05(x0是bias，就是上面的const)

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.951
Model:                            OLS   Adj. R-squared:                  0.945
Method:                 Least Squares   F-statistic:                     169.9
Date:                Thu, 11 Jan 2018   Prob (F-statistic):           1.34e-27
Time:                        16:49:15   Log-Likelihood:                -525.38
No. Observations:                  50   AIC:                             1063.
Df Residuals:                      44   BIC:                             1074.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       5.013e+04   6884.820      7.281      0.000    3.62e+04     6.4e+04
x1           198.7888   3371.007      0.059      0.953   -6595.030    6992.607
x2           -41.8870   3256.039     -0.013      0.990   -6604.003    6520.229
x3             0.8060      0.046     17.369      0.000       0.712       0.900
x4            -0.0270      0.052     -0.517      0.608      -0.132       0.078
x5             0.0270      0.017      1.574      0.123      -0.008       0.062
==============================================================================
Omnibus:                       14.782   Durbin-Watson:                   1.283
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               21.266
Skew:                          -0.948   Prob(JB):                     2.41e-05
Kurtosis:                       5.572   Cond. No.                     1.45e+06
==============================================================================
```

将x2删除，继续拟合模型，并查看其p值

```python
X_opt = X[:, [0, 1, 3, 4, 5]]
regressor_OLS = sm.OLS(endog=y, exog=X_opt).fit()
regressor_OLS.summary()
```

结果显示如下，可看出，x1的p值为0.940，是最大的，且大于0.05

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.951
Model:                            OLS   Adj. R-squared:                  0.946
Method:                 Least Squares   F-statistic:                     217.2
Date:                Thu, 11 Jan 2018   Prob (F-statistic):           8.49e-29
Time:                        16:52:24   Log-Likelihood:                -525.38
No. Observations:                  50   AIC:                             1061.
Df Residuals:                      45   BIC:                             1070.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       5.011e+04   6647.870      7.537      0.000    3.67e+04    6.35e+04
x1           220.1585   2900.536      0.076      0.940   -5621.821    6062.138
x2             0.8060      0.046     17.606      0.000       0.714       0.898
x3            -0.0270      0.052     -0.523      0.604      -0.131       0.077
x4             0.0270      0.017      1.592      0.118      -0.007       0.061
==============================================================================
Omnibus:                       14.758   Durbin-Watson:                   1.282
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               21.172
Skew:                          -0.948   Prob(JB):                     2.53e-05
Kurtosis:                       5.563   Cond. No.                     1.40e+06
==============================================================================
```

将x1删除，继续拟合模型

```
X_opt = X[:, [0, 3, 4, 5]]
regressor_OLS = sm.OLS(endog=y, exog=X_opt).fit()
regressor_OLS.summary()
```

结果显示如下，可看出，x2的p值为0.602，是最大的，且大于0.05

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.951
Model:                            OLS   Adj. R-squared:                  0.948
Method:                 Least Squares   F-statistic:                     296.0
Date:                Thu, 11 Jan 2018   Prob (F-statistic):           4.53e-30
Time:                        16:54:45   Log-Likelihood:                -525.39
No. Observations:                  50   AIC:                             1059.
Df Residuals:                      46   BIC:                             1066.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       5.012e+04   6572.353      7.626      0.000    3.69e+04    6.34e+04
x1             0.8057      0.045     17.846      0.000       0.715       0.897
x2            -0.0268      0.051     -0.526      0.602      -0.130       0.076
x3             0.0272      0.016      1.655      0.105      -0.006       0.060
==============================================================================
Omnibus:                       14.838   Durbin-Watson:                   1.282
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               21.442
Skew:                          -0.949   Prob(JB):                     2.21e-05
Kurtosis:                       5.586   Cond. No.                     1.40e+06
==============================================================================
```

将x2(就是原来的第4个变量)删除，继续拟合模型

结果显示如下，可看出，x2的p值为0.060，还是大于0.05

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.950
Model:                            OLS   Adj. R-squared:                  0.948
Method:                 Least Squares   F-statistic:                     450.8
Date:                Thu, 11 Jan 2018   Prob (F-statistic):           2.16e-31
Time:                        16:56:47   Log-Likelihood:                -525.54
No. Observations:                  50   AIC:                             1057.
Df Residuals:                      47   BIC:                             1063.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       4.698e+04   2689.933     17.464      0.000    4.16e+04    5.24e+04
x1             0.7966      0.041     19.266      0.000       0.713       0.880
x2             0.0299      0.016      1.927      0.060      -0.001       0.061
==============================================================================
Omnibus:                       14.677   Durbin-Watson:                   1.257
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               21.161
Skew:                          -0.939   Prob(JB):                     2.54e-05
Kurtosis:                       5.575   Cond. No.                     5.32e+05
==============================================================================
```

将x2(即原来的第5个变量)，然后继续拟合

```
X_opt = X[:, [0, 3]]
regressor_OLS = sm.OLS(endog=y, exog=X_opt).fit()
regressor_OLS.summary()
```

结果显示如下，现在没有一个变量的p值大于0.05了，所以这就是最优的模型。

可以发现，第3个变量(R&D Spend)对Profit的影响是最重要的，而且是唯一的预测变量(在SL=0.05的条件下)。

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.947
Model:                            OLS   Adj. R-squared:                  0.945
Method:                 Least Squares   F-statistic:                     849.8
Date:                Thu, 11 Jan 2018   Prob (F-statistic):           3.50e-32
Time:                        16:59:07   Log-Likelihood:                -527.44
No. Observations:                  50   AIC:                             1059.
Df Residuals:                      48   BIC:                             1063.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       4.903e+04   2537.897     19.320      0.000    4.39e+04    5.41e+04
x1             0.8543      0.029     29.151      0.000       0.795       0.913
==============================================================================
Omnibus:                       13.727   Durbin-Watson:                   1.116
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               18.536
Skew:                          -0.911   Prob(JB):                     9.44e-05
Kurtosis:                       5.361   Cond. No.                     1.65e+05
==============================================================================
```

关键词

> *back elimiation：反向淘汰，用于逐步回归法中淘汰对预测影响过小的变量，防止过多的无用变量对预测导致不良影响，以便防止过拟合*

> *dummy variable trap：虚拟变量陷阱，防止虚拟变量间产生的多重共线性（有多重共线性的数据集，使用回归分析会产生很大的问题）*

> *P-Value：P值，用来判定假设检验结果的一个参数，就是当原假设为真时所得到的样本观察结果或更极端结果出现的概率，如p值过小，会选择使用备择假设。在此例中，*
>
> *原假设：该变量的权值为0*
>
> *备择假设：该变量的权值不为0*

全部代码:

```
#Multiple Linear Regression

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('50_Startups.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values

#Encoding categorical data
#Encoding the Independent Variable
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelEncoder_X = LabelEncoder()
X[:,3] = labelEncoder_X.fit_transform(X[:,3])
onehotencoder = OneHotEncoder(categorical_features=[3])
X = onehotencoder.fit_transform(X).toarray()

#Avoiding the Dummy Variable Trap
X = X[:,1:]

# Splitting the dataset into the Training set and Test set
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 0)

#Fitting Multiple Linear Regression to the  Training set
from sklearn.linear_model import  LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

#Predicting the Test set results
y_pred = regressor.predict(X_test)

#Building the optimal model using Backward Elimination
import statsmodels.formula.api as sm
X = np.append(values=X,arr=np.ones((50,1)).astype(int),axis=1)
X_opt = X[:, [0,1,2,3,4,5]]

regressor_OLS = sm.OLS(endog=y, exog=X_opt).fit()
regressor_OLS.summary()
X_opt = X[:, [0, 1, 3, 4, 5]]
regressor_OLS = sm.OLS(endog=y, exog=X_opt).fit()
regressor_OLS.summary()

X_opt = X[:, [0, 3, 4, 5]]
```

```
regressor_OLS = sm.OLS(endog=y, exog=X_opt).fit()
regressor_OLS.summary()
X_opt = X[:, [0, 3, 5]]
regressor_OLS = sm.OLS(endog=y, exog=X_opt).fit()
regressor_OLS.summary()
X_opt = X[:, [0, 3]]
regressor_OLS = sm.OLS(endog=y, exog=X_opt).fit()
regressor_OLS.summary()
```

代码github地址：multiple_linear_regression.py