BACKWARD ELIMINATION METHOD(MANUAL)[¶](http://localhost:8888/notebooks/ML_data/multiple_linear_regression.ipynb#BACKWARD-ELIMINATION-METHOD(MANUAL)) + THE NOTEPAD FILE(AT THE END)

In [18]:



x

import statsmodels.api as sm

# remove one dummy variable to avoid the dummy variable trap(always use n-1 dummy variable)

x = X[:,1:]

print(x)

# add the intercept/constant in x

# we use np.ones to create a 50 rows 1 column numpy array to add in x

# we are using np.append to join these two numpy arrays

# numpy.append(arr,values,axis) parameters arr- the array ,values - value we want to append,

# axis - determines the axis of appending

x = np.append(arr = np.ones((50,1)).astype(int),values = x, axis = 1)

print('x after adding the intercept')

print(x)

[[0.0 1.0 165349.2 136897.8 471784.1]

[0.0 0.0 162597.7 151377.59 443898.53]

[1.0 0.0 153441.51 101145.55 407934.54]

[0.0 1.0 144372.41 118671.85 383199.62]

[1.0 0.0 142107.34 91391.77 366168.42]

[0.0 1.0 131876.9 99814.71 362861.36]

[0.0 0.0 134615.46 147198.87 127716.82]

[1.0 0.0 130298.13 145530.06 323876.68]

[0.0 1.0 120542.52 148718.95 311613.29]

[0.0 0.0 123334.88 108679.17 304981.62]

[1.0 0.0 101913.08 110594.11 229160.95]

[0.0 0.0 100671.96 91790.61 249744.55]

[1.0 0.0 93863.75 127320.38 249839.44]

[0.0 0.0 91992.39 135495.07 252664.93]

[1.0 0.0 119943.24 156547.42 256512.92]

[0.0 1.0 114523.61 122616.84 261776.23]

[0.0 0.0 78013.11 121597.55 264346.06]

[0.0 1.0 94657.16 145077.58 282574.31]

[1.0 0.0 91749.16 114175.79 294919.57]

[0.0 1.0 86419.7 153514.11 0.0]

[0.0 0.0 76253.86 113867.3 298664.47]

[0.0 1.0 78389.47 153773.43 299737.29]

[1.0 0.0 73994.56 122782.75 303319.26]

[1.0 0.0 67532.53 105751.03 304768.73]

[0.0 1.0 77044.01 99281.34 140574.81]

[0.0 0.0 64664.71 139553.16 137962.62]

[1.0 0.0 75328.87 144135.98 134050.07]

[0.0 1.0 72107.6 127864.55 353183.81]

[1.0 0.0 66051.52 182645.56 118148.2]

[0.0 1.0 65605.48 153032.06 107138.38]

[1.0 0.0 61994.48 115641.28 91131.24]

[0.0 1.0 61136.38 152701.92 88218.23]

[0.0 0.0 63408.86 129219.61 46085.25]

[1.0 0.0 55493.95 103057.49 214634.81]

[0.0 0.0 46426.07 157693.92 210797.67]

[0.0 1.0 46014.02 85047.44 205517.64]

[1.0 0.0 28663.76 127056.21 201126.82]

[0.0 0.0 44069.95 51283.14 197029.42]

[0.0 1.0 20229.59 65947.93 185265.1]

[0.0 0.0 38558.51 82982.09 174999.3]

[0.0 0.0 28754.33 118546.05 172795.67]

[1.0 0.0 27892.92 84710.77 164470.71]

[0.0 0.0 23640.93 96189.63 148001.11]

[0.0 1.0 15505.73 127382.3 35534.17]

[0.0 0.0 22177.74 154806.14 28334.72]

[0.0 1.0 1000.23 124153.04 1903.93]

[1.0 0.0 1315.46 115816.21 297114.46]

[0.0 0.0 0.0 135426.92 0.0]

[0.0 1.0 542.05 51743.15 0.0]

[0.0 0.0 0.0 116983.8 45173.06]]

x after adding the intercept

[[1 0.0 1.0 165349.2 136897.8 471784.1]

[1 0.0 0.0 162597.7 151377.59 443898.53]

[1 1.0 0.0 153441.51 101145.55 407934.54]

[1 0.0 1.0 144372.41 118671.85 383199.62]

[1 1.0 0.0 142107.34 91391.77 366168.42]

[1 0.0 1.0 131876.9 99814.71 362861.36]

[1 0.0 0.0 134615.46 147198.87 127716.82]

[1 1.0 0.0 130298.13 145530.06 323876.68]

[1 0.0 1.0 120542.52 148718.95 311613.29]

[1 0.0 0.0 123334.88 108679.17 304981.62]

[1 1.0 0.0 101913.08 110594.11 229160.95]

[1 0.0 0.0 100671.96 91790.61 249744.55]

[1 1.0 0.0 93863.75 127320.38 249839.44]

[1 0.0 0.0 91992.39 135495.07 252664.93]

[1 1.0 0.0 119943.24 156547.42 256512.92]

[1 0.0 1.0 114523.61 122616.84 261776.23]

[1 0.0 0.0 78013.11 121597.55 264346.06]

[1 0.0 1.0 94657.16 145077.58 282574.31]

[1 1.0 0.0 91749.16 114175.79 294919.57]

[1 0.0 1.0 86419.7 153514.11 0.0]

[1 0.0 0.0 76253.86 113867.3 298664.47]

[1 0.0 1.0 78389.47 153773.43 299737.29]

[1 1.0 0.0 73994.56 122782.75 303319.26]

[1 1.0 0.0 67532.53 105751.03 304768.73]

[1 0.0 1.0 77044.01 99281.34 140574.81]

[1 0.0 0.0 64664.71 139553.16 137962.62]

[1 1.0 0.0 75328.87 144135.98 134050.07]

[1 0.0 1.0 72107.6 127864.55 353183.81]

[1 1.0 0.0 66051.52 182645.56 118148.2]

[1 0.0 1.0 65605.48 153032.06 107138.38]

[1 1.0 0.0 61994.48 115641.28 91131.24]

[1 0.0 1.0 61136.38 152701.92 88218.23]

[1 0.0 0.0 63408.86 129219.61 46085.25]

[1 1.0 0.0 55493.95 103057.49 214634.81]

[1 0.0 0.0 46426.07 157693.92 210797.67]

[1 0.0 1.0 46014.02 85047.44 205517.64]

[1 1.0 0.0 28663.76 127056.21 201126.82]

[1 0.0 0.0 44069.95 51283.14 197029.42]

[1 0.0 1.0 20229.59 65947.93 185265.1]

[1 0.0 0.0 38558.51 82982.09 174999.3]

[1 0.0 0.0 28754.33 118546.05 172795.67]

[1 1.0 0.0 27892.92 84710.77 164470.71]

[1 0.0 0.0 23640.93 96189.63 148001.11]

[1 0.0 1.0 15505.73 127382.3 35534.17]

[1 0.0 0.0 22177.74 154806.14 28334.72]

[1 0.0 1.0 1000.23 124153.04 1903.93]

[1 1.0 0.0 1315.46 115816.21 297114.46]

[1 0.0 0.0 0.0 135426.92 0.0]

[1 0.0 1.0 542.05 51743.15 0.0]

[1 0.0 0.0 0.0 116983.8 45173.06]]

. . .

In [20]:



x

# create my optimal matrix with all feature(creating a copy of x)

x\_opt = np.array(x[:,[0,1,2,3,4,5]],dtype = float)

# we created and fitted the model using statsmodels module of python which is using ordinary least qsquare method

regressor = sm.OLS(endog = y, exog = x\_opt).fit()

# getting the summary of the model

regressor.summary()

Out[20]:

|  |  |  |  |
| --- | --- | --- | --- |
| 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: | Mon, 18 May 2020 | Prob (F-statistic): | 1.34e-27 |
| Time: | 19:12:06 | 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 |

Warnings:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The condition number is large, 1.45e+06. This might indicate that there are  
strong multicollinearity or other numerical problems.

. . .

In [21]:



x

# removing the first feature i.e. d2 with p = 0.99 > SL = 0.05

x\_opt = np.array(x[:,[0,1,3,4,5]],dtype = float)

# we will refit the model with the remaining features

regressor = sm.OLS(endog = y, exog = x\_opt).fit()

# getting the summary of the model

regressor.summary()

Out[21]:

|  |  |  |  |
| --- | --- | --- | --- |
| 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: | Mon, 18 May 2020 | Prob (F-statistic): | 8.49e-29 |
| Time: | 19:18:10 | 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 |

Warnings:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The condition number is large, 1.4e+06. This might indicate that there are  
strong multicollinearity or other numerical problems.

. . .

In [22]:



x

# removing the second feature i.e. d2 with p = 0.99 > SL = 0.05

x\_opt = np.array(x[:,[0,3,4,5]],dtype = float)

# we will refit the model with the remaining features

regressor = sm.OLS(endog = y, exog = x\_opt).fit()

# getting the summary of the model

regressor.summary()

Out[22]:

|  |  |  |  |
| --- | --- | --- | --- |
| 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: | Mon, 18 May 2020 | Prob (F-statistic): | 4.53e-30 |
| Time: | 19:20:30 | 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 |

Warnings:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The condition number is large, 1.4e+06. This might indicate that there are  
strong multicollinearity or other numerical problems.

. . .

In [23]:



x

# removing the third feature i.e. d2 with p = 0.99 > SL = 0.05

x\_opt = np.array(x[:,[0,3,5]],dtype = float)

# we will refit the model with the remaining features

regressor = sm.OLS(endog = y, exog = x\_opt).fit()

# getting the summary of the model

regressor.summary()

Out[23]:

|  |  |  |  |
| --- | --- | --- | --- |
| 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: | Mon, 18 May 2020 | Prob (F-statistic): | 2.16e-31 |
| Time: | 19:22:23 | 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 |

Warnings:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The condition number is large, 5.32e+05. This might indicate that there are  
strong multicollinearity or other numerical problems.

. . .

In [24]:



# removing the Fourth feature i.e. d2 with p = 0.99 > SL = 0.05

x\_opt = np.array(x[:,[0,3]],dtype = float)

# we will refit the model with the remaining features

regressor = sm.OLS(endog = y, exog = x\_opt).fit()

# getting the summary of the model

regressor.summary()

​

#p-value for both = 0.00 < SL = 0.05

#so we are going to stop

#and we have successfully eliminated the insignificant features

Out[24]:

|  |  |  |  |
| --- | --- | --- | --- |
| 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: | Mon, 18 May 2020 | Prob (F-statistic): | 3.50e-32 |
| Time: | 19:23:59 | 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 |

Warnings:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The condition number is large, 1.65e+05. This might indicate that there are  
strong multicollinearity or other numerical problems.

NOTEPAD FILE(ASSIGNMENT AT THE END)

if you are having xlrd error

pip install xlrd

in programming there is not a hard and fast rule to solve a problem in a single way.

we can try to remember the logic you can try to formulate a solution(internet helps in finding the library/module/class/function)

my laptop is a bit old

some ML models are heavy duty(for those models i cannot run the model adn webinar togrther)

classification Model(huge)

every time i tried to run , the classification in the session my laptop

scikit-learn.org - API

for those my vice lagging/breaking

please log-out anf log-in i will wait for few seconds

there are two ways to do

feature elimination

feature selection

feature elimination - backward elimination(easiest to understand and work-with)

feature selection - PCA(4 week)

we need statsmodels.api

1-we fit the model with all the features(also includes dummy variables and the intercept)

2- we calculate the p-value for all the features

3-choose the feature with the maximum p-value

compare the max p-value with SL(0.05)

if the max p-value > SL we remove that feature

4 - we go back and again refit our model with the remaining feature

5 - we are going to do this step until the feature with p-value < SL remians

(model is finished)

indices for the x\_opt (after step-1)

constant - x0

d1 - 1 (x1)

d2 - 2 (x2)(max p-value = 0.99)

r&d - 3 (x3)

admin - 4 (x4)

marketing - 5 (x4)

after first elimination

constant - 0 (x0)

d1 - 1 (x1)

r&d - 3 (x2)

admin - 4 (x3)

marketing - 5 (x4)

after second elimination

constant - 0 (x0)

r&d - 3 (x1)

admin - 4 (x2)

marketing - 5 (x3)

after third elimination

constant - 0 (x0)

r&d - 3 (x1)

marketing - 5 (x2)

after the fourth elimination

constant - 0 (x0)

r&d - 3 (x1)

p-value for both = 0.00 < SL = 0.05

so we are going to stop

and we have successfully eliminated the insignificant features

Assignment for today is to:

try to automate BACKWARD ELIMINATION