

## Multiple Linear Regression

A wealthy investor gave me a list of 50 anonymous businesses. He desired to know which factor was most important in knowing which business was best to invest in (information shown in the datasheet *50\_Startups.csv*).

The datasheet came with the following information / factors.

- R&D spending's
- Administration spending's
- Marketing spending's
- State (California, New York, or Florida)
- Profit

Index	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349	136898	471784	New York	192262
1	162598	151378	443899	California	191792
2	153442	101146	407935	Florida	191050
3	144372	118672	383200	New York	182902
4	142107	91392	366168	Florida	166188
5	131877	99815	362861	New York	156991

In order to solve this problem, I decided to apply the machine learning method multiple linear regression (MLR).

The profit became our dependent variable (as this is what the investor was interested in) while the other four were independent variables. I split the table into X (independent variables) and Y (dependent variable). I then took the *state* and transformed it from a categorical variable into a linear variable. I did this by creating 3 dummy variables where,

[0, 0, 1] = New York

[1, 0, 0] = California

[0, 1, 0] = Florida

	0	1	2
0	0	0	1
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

I then removed the first dummy variable in order to bypass dummy variable trap. Out tables appeared as such (X on left, Y on right).

	0	1	2	3	4		0
0	0.000	1.000	165349.200	136897.800	471784.100	0	192261.830
1	0.000	0.000	162597.700	151377.590	443898.530	1	191792.060
2	1.000	0.000	153441.510	101145.550	407934.540	2	191050.390
3	0.000	1.000	144372.410	118671.850	383199.620	3	182901.990
4	1.000	0.000	142107.340	91391.770	366168.420	4	166187.940
5	0.000	1.000	131876.900	99814.710	362861.360	5	156991.120
6	0.000	0.000	134615.460	147198.870	127716.820	6	156122.510
7	1.000	0.000	130298.130	145530.060	323876.680	7	155752.600
8	0.000	1.000	120542.520	148718.950	311613.290	8	152211.770
9	0.000	0.000	123334.880	108679.170	304981.620	9	149759.960

I then split the table into training (80%) and test (20%) sets. Then using linear regression, I predict the test set results. After this I also added a column of 1's in order to account for the constant to the independent variables (X). The final independent dataset appeared as follows.

	0	1	2	3	4	5
	Row of 1's (constant)	Dummy Variable 1	Dummy Variable 2	R&D	Administration	Marketing
	0	1	2	3	4	5
0	1.000	0.000	1.000	165349.200	136897.800	471784.100
1	1.000	0.000	0.000	162597.700	151377.590	443898.530
2	1.000	1.000	0.000	153441.510	101145.550	407934.540
3	1.000	0.000	1.000	144372.410	118671.850	383199.620
4	1.000	1.000	0.000	142107.340	91391.770	366168.420
5	1.000	0.000	1.000	131876.900	99814.710	362861.360
6	1.000	0.000	0.000	134615.460	147198.870	127716.820
7	1.000	1.000	0.000	130298.130	145530.060	323876.680
8	1.000	0.000	1.000	120542.520	148718.950	311613.290
9	1.000	0.000	0.000	123334.880	108679.170	304981.620

I used the Backward Elimination method to find the optimal model (see output below). I used a significance level of 5%. This resulted in the removing of the following independent variables: Dummy Variable 2 (2), Dummy Variable 1 (1), Administration (4), Marketing (5). There could be consideration for keeping Marketing (5) for its significance level is at 6%. One argument for keeping Marketing (5) is that the  $R^2$  and adj- $R^2$  value goes down which means that our model has become less accurate.

Through this we found out that R&D is the most important factor in regards to predicting the profit of the company. Second, because the significance level of marketing is so close to our set significance level it would also be advisable to include marketing in predicting the profit of the company.

```

...: # Optimal matrix of features (Significant level is 5%)
...: X_opt = X[:, [0, 1, 2, 3, 4, 5]]
...: lr_ols = sm.OLS(endog = Y, exog = X_opt).fit()
...: lr_ols.summary()
Out[17]:
<class 'statsmodels.iolib.summary.Summary'>
=====
                        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:                Wed, 02 Aug 2017  Prob (F-statistic):        1.34e-27
Time:                08:01:46      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
=====

```

```
In [4]: X_opt = X[:, [0, 1, 3, 4, 5]]
....: lr_ols = sm.OLS(endog = Y, exog = X_opt).fit()
....: lr_ols.summary()
Out[4]:
<class 'statsmodels.iolib.summary.Summary'>
=====
                        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:                Wed, 02 Aug 2017  Prob (F-statistic):    8.49e-29
Time:                09:34:26   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
=====
```

```
In [5]: X_opt = X[:, [0, 3, 4, 5]]
....: lr_ols = sm.OLS(endog = Y, exog = X_opt).fit()
....: lr_ols.summary()
Out[5]:
<class 'statsmodels.iolib.summary.Summary'>
=====
                        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:                Wed, 02 Aug 2017  Prob (F-statistic):    4.53e-30
Time:                09:35:07   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
=====
```

```
In [6]: X_opt = X[:, [0, 3, 5]]
....: lr_ols = sm.OLS(endog = Y, exog = X_opt).fit()
....: lr_ols.summary()
Out[6]:
<class 'statsmodels.iolib.summary.Summary'>
=====
                        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:                   Wed, 02 Aug 2017    Prob (F-statistic):  2.16e-31
Time:                   09:37:06    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
=====
```

```
In [7]: X_opt = X[:, [0, 3]]
....: lr_ols = sm.OLS(endog = Y, exog = X_opt).fit()
....: lr_ols.summary()
Out[7]:
<class 'statsmodels.iolib.summary.Summary'>
=====
                        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:                   Wed, 02 Aug 2017    Prob (F-statistic):  3.50e-32
Time:                   09:37:51    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
=====
```