# miniproject1

February 11, 2022

[1]:

```
# %load ../standard_import.txt
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
import seaborn as sns

from sklearn.preprocessing import scale
import sklearn.linear_model as skl_lm
from sklearn.metrics import mean_squared_error, r2_score
import statsmodels.api as sm
import statsmodels.formula.api as smf
from sklearn.linear_model import LinearRegression

%matplotlib inline
plt.style.use('seaborn-white')
```

[2]:

```
advertising = pd.read_csv('Advertising.csv', usecols=[1,2,3,4])
advertising.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   TV         200 non-null    float64
 1   radio      200 non-null    float64
 2   newspaper  200 non-null    float64
 3   sales      200 non-null    float64
dtypes: float64(4)
memory usage: 6.4 KB
```

[3]:

```
print(advertising)
```

```
        TV  radio  newspaper  sales
0    230.1   37.8       69.2   22.1
1     44.5   39.3       45.1   10.4
2     17.2   45.9       69.3    9.3
3    151.5   41.3       58.5   18.5
```

```
4    180.8   10.8       58.4   12.9
..     …      …          …      …
195   38.2    3.7       13.8    7.6
196   94.2    4.9        8.1    9.7
197  177.0    9.3        6.4   12.8
198  283.6   42.0       66.2   25.5
199  232.1    8.6        8.7   13.4

[200 rows x 4 columns]
```

[4]:
```
x_1 = advertising.TV
x_2 = advertising.radio
print(x_1,x_2)
```

```
0      230.1
1       44.5
2       17.2
3      151.5
4      180.8
       …
195     38.2
196     94.2
197    177.0
198    283.6
199    232.1
Name: TV, Length: 200, dtype: float64 0      37.8
1      39.3
2      45.9
3      41.3
4      10.8
       …
195     3.7
196     4.9
197     9.3
198    42.0
199     8.6
Name: radio, Length: 200, dtype: float64
```

[5]:
```
advertising["TV*radio"] = advertising["TV"] * advertising["radio"]
```

[6]:
```
print(advertising)
```

```
        TV  radio  newspaper  sales  TV*radio
0    230.1   37.8       69.2   22.1   8697.78
1     44.5   39.3       45.1   10.4   1748.85
2     17.2   45.9       69.3    9.3    789.48
3    151.5   41.3       58.5   18.5   6256.95
4    180.8   10.8       58.4   12.9   1952.64
..     …      …          …      …        …
```

```
195   38.2    3.7        13.8    7.6    141.34
196   94.2    4.9         8.1    9.7    461.58
197  177.0    9.3         6.4   12.8   1646.10
198  283.6   42.0        66.2   25.5  11911.20
199  232.1    8.6         8.7   13.4   1996.06

[200 rows x 5 columns]
```

[7]:
```
x = advertising[["TV","radio","TV*radio"]]
```

[8]:
```
print(x)
```

```
        TV  radio  TV*radio
0    230.1   37.8   8697.78
1     44.5   39.3   1748.85
2     17.2   45.9    789.48
3    151.5   41.3   6256.95
4    180.8   10.8   1952.64
..     …     …        …
195   38.2    3.7    141.34
196   94.2    4.9    461.58
197  177.0    9.3   1646.10
198  283.6   42.0  11911.20
199  232.1    8.6   1996.06

[200 rows x 3 columns]
```

[9]:
```
x_3 = x_1*x_2
print(x_3)
```

```
0       8697.78
1       1748.85
2        789.48
3       6256.95
4       1952.64
         …
195      141.34
196      461.58
197     1646.10
198    11911.20
199     1996.06
Length: 200, dtype: float64
```

[10]:
```
type(x_3)
```

[10]:
```
pandas.core.series.Series
```

[11]:
```
x = pd.concat([x_1, x_2, x_3], axis=1)
print(x)
```

```
        TV  radio         0
0    230.1   37.8   8697.78
1     44.5   39.3   1748.85
2     17.2   45.9    789.48
3    151.5   41.3   6256.95
4    180.8   10.8   1952.64
..     …     …        …
195   38.2    3.7    141.34
196   94.2    4.9    461.58
197  177.0    9.3   1646.10
198  283.6   42.0  11911.20
199  232.1    8.6   1996.06

[200 rows x 3 columns]
```

[12]:
```python
y = advertising.sales
print(y)
```

```
0      22.1
1      10.4
2       9.3
3      18.5
4      12.9
       …
195     7.6
196     9.7
197    12.8
198    25.5
199    13.4
Name: sales, Length: 200, dtype: float64
```

[13]:
```python
est = smf.ols('sales ~ TV + radio + TV*radio', advertising).fit()
est.summary().tables[1]
```

[13]:
```
<class 'statsmodels.iolib.table.SimpleTable'>
```

[14]:
```python
reg = LinearRegression().fit(x, y)
```

[15]:
```python
reg.score(x, y)
```

[15]:
```
0.9677905498482523
```

[16]:
```python
reg.coef_
```

[16]:
```
array([0.01910107, 0.02886034, 0.00108649])
```

[17]:
```python
reg.intercept_
```

[17]:
```
6.750220203075117
```

### 0.0.1 The intercept and coefficients of TV, Radio and Radio*TV are same with they calculated by smf toolbox