Assignment 3

Simple and Multiple Linear Regression pt2

Author: Samuel Fredric Berg

Student ID: sb224sc

Date: 2026-02-09

Course: Machine Learning 4DT905

Conceptual

$$Y = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \hat{\beta}_3 X_3 + \hat{\beta}_4 X_1 X_2 + \hat{\beta}_5 X_1 X_3$$

$$\hat{\beta}_0 = 50(Intercept)$$

$$\hat{\beta}_1 = 20(GPA)$$

$$\hat{\beta}_2 = 0.07(IQ)$$

$$\hat{\beta}_3 = 35(Level)$$

$$\hat{\beta}_4 = 0.01(GPA \cdot IQ)$$

$$\hat{\beta}_5 = -10(GPA \cdot Level)$$

$$X_3 = 1 \text{ for College, } 0 \text{ for High School}$$

1.

$$Y_c = 50 + 20X_1 + 0.07X_2 + 35 + 0.01X_1X_2 - 10X_1$$

$$Y_h = 50 + 20X_1 + 0.07X_2 + 0.01X_1X_2$$

$$Y_c - Y_h = 35 - 10X_1$$

$$35 - 10X_1 = 0 \implies X_1 = 3.5$$

Thue, when GPA > 3.5 High School graduates earn more than College graduates.

Answer: $\boxed{iii}$

2.

$$X_1 = 4.0$$

$$X_2 = 110$$

$$X_3 = 1$$

$$Y = 50 + 20(4.0) + 0.07(110) + 35 + 0.01(4.0)(110) - 10(4.0)$$

$$Y = 137.1$$

Answer: $\boxed{\$137,100}$

3. False. The magnitude of a coefficient does not indicate statistical importance. To determine statistical importance we need to look at the p-values associated with that coefficient, not just its absolute value. In the presented case, the units of predictor X_2 (IQ) are generally > 100. A small coeffeicient for the $X_2 \cdot X_1$ term might still result in a large contribution to the model and be highly statistically significant.

Answer: $\boxed{False}$

Practical

Imports

In [1]:
```
import pandas as pd
import statsmodels.api as sm
import numpy as np
```

Load data

In [2]:
```
df = pd.read_csv("../data/Boston.csv", index_col=0)
```

In [3]:
```
X = df[["lstat", "rm", "nox", "dis", "ptratio"]]
Y = df["medv"]
X = sm.add_constant(X)
model1 = sm.OLS(Y, X).fit()

print(model1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.708
Model:                            OLS   Adj. R-squared:                  0.705
Method:                 Least Squares   F-statistic:                     242.6
Date:                Mon, 09 Feb 2026   Prob (F-statistic):         3.67e-131
Time:                        06:38:55   Log-Likelihood:                -1528.7
No. Observations:                 506   AIC:                             3069.
Df Residuals:                     500   BIC:                             3095.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         37.4992      4.613      8.129      0.000      28.436      46.562
lstat         -0.5811      0.048    -12.122      0.000      -0.675      -0.487
rm             4.1633      0.412     10.104      0.000       3.354       4.973
nox          -17.9966      3.261     -5.519      0.000     -24.403     -11.590
dis           -1.1847      0.168     -7.034      0.000      -1.516      -0.854
ptratio       -1.0458      0.114     -9.212      0.000      -1.269      -0.823
==============================================================================
Omnibus:                      187.456   Durbin-Watson:                   0.971
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              885.498
Skew:                           1.584   Prob(JB):                    5.21e-193
Kurtosis:                       8.654   Cond. No.                         545.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
```

In [4]:

```
df["lstat_rm"] = df["lstat"] * df["rm"]
X = df[["lstat", "rm", "lstat_rm", "nox", "dis", "ptratio"]]
X = sm.add_constant(X)
model2 = sm.OLS(Y, X).fit()

print(model2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.778
Model:                            OLS   Adj. R-squared:                  0.775
Method:                 Least Squares   F-statistic:                     290.8
Date:                Mon, 09 Feb 2026   Prob (F-statistic):          2.48e-159
Time:                        06:38:55   Log-Likelihood:                -1459.9
No. Observations:                 506   AIC:                             2934.
Df Residuals:                     499   BIC:                             2963.
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          3.1518      4.880      0.646      0.519      -6.435      12.739
lstat          1.8115      0.196      9.237      0.000       1.426       2.197
rm             8.3344      0.491     16.971      0.000       7.370       9.299
lstat_rm      -0.4185      0.034    -12.488      0.000      -0.484      -0.353
nox          -12.3651      2.885     -4.286      0.000     -18.033      -6.697
dis           -1.0184      0.148     -6.893      0.000      -1.309      -0.728
ptratio       -0.7152      0.103     -6.967      0.000      -0.917      -0.514
==============================================================================
Omnibus:                      246.928   Durbin-Watson:                   1.079
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2792.613
Skew:                           1.836   Prob(JB):                         0.00
Kurtosis:                      13.908   Cond. No.                     2.36e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The condition number is large, 2.36e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Interpretation of results

By utilizing `lstat` , `rm` , `nox` , `dis` and `ptratio` columns, the model achieves an R-squared value of (0.705), but by just adding the interaction between `lstat` and `rm` the R-squared value increases to (0.775). This indicates that the interaction between `lstat` and `rm` contributes a better prediction of `medv` than just using the individual predictors alone.

Confidence Intervals: Examining the confidence intervals for the coefficients in model1 and model2, we should check whether they include zero (indicating significance) and their width (indicating precision of the estimate). Coefficients with confidence intervals that do not include zero are statistically significant predictors of the target variable.

Correlation and Multicollinearity: Before building the models, it would be beneficial to examine the correlation matrix to identify: (1) which predictors are most correlated with the target variable `medv` , and (2) whether there is multicollinearity between predictors (high correlation between predictor variables). High multicollinearity can affect the stability and interpretability of coefficient estimates.

Adding non-linear term

In [5]:
```
df["lstat_rm_squared"] = df["lstat_rm"] ** 2
X = df[["lstat", "rm", "lstat_rm", "lstat_rm_squared", "nox", "dis", "ptratio"]]
X = sm.add_constant(X)
model3 = sm.OLS(Y, X).fit()

print(model3.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.781
Model:                            OLS   Adj. R-squared:                  0.778
Method:                 Least Squares   F-statistic:                     253.9
Date:                Mon, 09 Feb 2026   Prob (F-statistic):          8.05e-160
Time:                        06:38:55   Log-Likelihood:                -1455.8
No. Observations:                 506   AIC:                             2928.
Df Residuals:                     498   BIC:                             2961.
Df Model:                           7
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const               10.5522      5.499      1.919      0.056      -0.253      21.357
lstat                1.5468      0.216      7.167      0.000       1.123       1.971
rm                   7.6004      0.552     13.777      0.000       6.516       8.684
lstat_rm            -0.4468      0.035    -12.864      0.000      -0.515      -0.379
lstat_rm_squared     0.0004      0.000      2.845      0.005       0.000       0.001
nox                -12.2898      2.865     -4.290      0.000     -17.918      -6.662
dis                 -1.0641      0.148     -7.209      0.000      -1.354      -0.774
ptratio             -0.7112      0.102     -6.977      0.000      -0.912      -0.511
==============================================================================
Omnibus:                      217.415   Durbin-Watson:                   1.059
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2007.945
Skew:                           1.622   Prob(JB):                         0.00
Kurtosis:                      12.204   Cond. No.                     3.02e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The condition number is large, 3.02e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Preform ANOVA

In [6]:
```
ANOVA_results = sm.stats.anova_lm(model2, model3)
print(ANOVA_results)
```

```
   df_resid          ssr  df_diff     ss_diff         F    Pr(>F)
0     499.0  9500.381881      0.0         NaN       NaN       NaN
1     498.0  9348.435955      1.0  151.945925  8.094303  0.004623
```

Conclusion from ANOVA

Hypothesis Test: The ANOVA F-test compares model2 (with interaction term) versus model3 (with interaction and non-linear term).

- Null Hypothesis (H0): The simpler model (model2) is sufficient.
- Alternative Hypothesis (H1): The more complex model (model3) provides a significantly better fit.

Result: With a p-value of 0.004 (< 0.05), we reject the null hypothesis. The ANOVA test indicates that model3 (with the squared interaction term) provides a statistically significantly better fit than model2. The F-statistic measures the improvement in model fit relative to the increase in model complexity. The significant p-value suggests that adding the non-linear term meaningfully improves the model's ability to explain variance in the target variable.

Add polynomial

In [7]:
```
for exp in range(2, 6):
    df[f"lstat_poly_{exp}"] = df["lstat"] ** exp

X = df[
    [
        "lstat",
        "rm",
        "lstat_rm",
        "lstat_poly_2",
        "lstat_poly_3",
        "lstat_poly_4",
        "lstat_poly_5",
        "nox",
        "dis",
        "ptratio",
    ]
]
X = sm.add_constant(X)
model4 = sm.OLS(Y, X).fit()

print(model4.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.792
Model:                            OLS   Adj. R-squared:                  0.787
Method:                 Least Squares   F-statistic:                     188.0
Date:                Mon, 09 Feb 2026   Prob (F-statistic):         1.80e-161
Time:                        06:38:55   Log-Likelihood:                -1443.5
No. Observations:                 506   AIC:                             2909.
Df Residuals:                     495   BIC:                             2956.
Df Model:                          10
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const           33.9246      7.663      4.427      0.000      18.869      48.981
lstat           -5.6422      1.426     -3.957      0.000      -8.444      -2.841
rm               6.5291      0.711      9.183      0.000       5.132       7.926
lstat_rm        -0.3055      0.052     -5.878      0.000      -0.408      -0.203
lstat_poly_2     0.8633      0.187      4.622      0.000       0.496       1.230
lstat_poly_3    -0.0495      0.012     -4.153      0.000      -0.073      -0.026
lstat_poly_4     0.0013      0.000      3.795      0.000       0.001       0.002
lstat_poly_5 -1.279e-05   3.64e-06     -3.514      0.000   -1.99e-05   -5.64e-06
nox            -13.7513      2.823     -4.871      0.000     -19.298      -8.204
dis             -1.0326      0.145     -7.127      0.000      -1.317      -0.748
ptratio         -0.7407      0.101     -7.324      0.000      -0.939      -0.542
==============================================================================
Omnibus:                      232.049   Durbin-Watson:                   1.116
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2275.267
Skew:                           1.742   Prob(JB):                         0.00
Kurtosis:                      12.787   Cond. No.                     3.38e+08
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The condition number is large, 3.38e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [8]:
```
ANOVA_results = sm.stats.anova_lm(model2, model4)
print(ANOVA_results)
```

```
   df_resid          ssr  df_diff     ss_diff         F    Pr(>F)
0     499.0  9500.381881      0.0         NaN       NaN       NaN
1     495.0  8903.772453      4.0  596.609428  8.292038  0.000002
```

Conclusion from ANOVA

Hypothesis Test: The ANOVA F-test compares model2 (with interaction term) versus model4 (with interaction and polynomial terms).

- Null Hypothesis (H0): The simpler model (model2) is sufficient.
- Alternative Hypothesis (H1): The more complex model (model4) provides a significantly better fit.

Result: With a very low p-value (0.000002 << 0.05), we reject the null hypothesis. The

ANOVA test indicates that model4 (with polynomial terms) provides a statistically significantly better fit than model2. However, it's important to note that high-degree polynomials increase the risk of overfitting to the training data, which can lead to poor generalization on new data. Cross-validation should be considered to assess true predictive performance.

In [9]:

```
df["log_rm"] = np.log(df["rm"])

X = df[
    [
        "lstat",
        "lstat_poly_2",
        "lstat_poly_3",
        "lstat_poly_4",
        "lstat_poly_5",
        "rm",
        "log_rm",
        "nox",
        "dis",
        "ptratio",
    ]
]
X = sm.add_constant(X)
model5 = sm.OLS(Y, X).fit()

print(model5.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.804
Model:                            OLS   Adj. R-squared:                  0.800
Method:                 Least Squares   F-statistic:                     202.6
Date:                Mon, 09 Feb 2026   Prob (F-statistic):          7.10e-168
Time:                        06:38:55   Log-Likelihood:                -1428.4
No. Observations:                 506   AIC:                             2879.
Df Residuals:                     495   BIC:                             2925.
Df Model:                          10
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const          172.9866     13.954     12.397      0.000     145.571     200.402
lstat           -8.5527      1.227     -6.969      0.000     -10.964      -6.141
lstat_poly_2     1.0064      0.178      5.654      0.000       0.657       1.356
lstat_poly_3    -0.0582      0.011     -5.087      0.000      -0.081      -0.036
lstat_poly_4     0.0015      0.000      4.672      0.000       0.001       0.002
lstat_poly_5 -1.521e-05   3.52e-06     -4.323      0.000    -2.21e-05    -8.3e-06
rm              25.1967      2.732      9.224      0.000      19.830      30.564
log_rm        -137.4038     16.761     -8.198      0.000    -170.336    -104.472
nox            -16.6408      2.734     -6.087      0.000     -22.012     -11.270
dis             -0.9709      0.141     -6.885      0.000      -1.248      -0.694
ptratio         -0.7843      0.097     -8.116      0.000      -0.974      -0.594
==============================================================================
Omnibus:                      221.958   Durbin-Watson:                   1.064
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2718.500
Skew:                           1.567   Prob(JB):                         0.00
Kurtosis:                      13.914   Cond. No.                     9.63e+08
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The condition number is large, 9.63e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [10]:

```python
ANOVA_results = sm.stats.anova_lm(model2, model5)
print(ANOVA_results)
```

```
   df_resid          ssr  df_diff      ss_diff          F        Pr(>F)
0     499.0  9500.381881      0.0          NaN        NaN           NaN
1     495.0  8386.756361      4.0  1113.62552  16.431997  1.190966e-12
```

Conclusion from ANOVA

Hypothesis Test: The ANOVA F-test compares model2 (with interaction term) versus model5 (with interaction, polynomial, and logarithmic terms).

- Null Hypothesis (H0): The simpler model (model2) is sufficient.
- Alternative Hypothesis (H1): The more complex model (model5) provides a significantly better fit.

Result: With a very low p-value (much less than 0.05), we reject the null hypothesis. The

ANOVA test indicates that model5 provides a statistically significantly better fit than model2. The increase in R-squared value demonstrates improved explanatory power. The combination of polynomial and logarithmic transformations captures both polynomial trends and logarithmic relationships in the data.

Load data 2

In [11]:
```
df2 = pd.read_csv("../data/Carseats.csv", index_col=0)
print(df2.describe(), "\n")
print(df2["ShelveLoc"].value_counts(), "\n")
print(df2["Urban"].value_counts(), "\n")
print(df2["US"].value_counts())
```

```
            Sales   CompPrice      Income  Advertising  Population  \
count  400.000000  400.000000  400.000000   400.000000  400.000000
mean     7.496325  124.975000   68.657500     6.635000  264.840000
std      2.824115   15.334512   27.986037     6.650364  147.376436
min      0.000000   77.000000   21.000000     0.000000   10.000000
25%      5.390000  115.000000   42.750000     0.000000  139.000000
50%      7.490000  125.000000   69.000000     5.000000  272.000000
75%      9.320000  135.000000   91.000000    12.000000  398.500000
max     16.270000  175.000000  120.000000    29.000000  509.000000

            Price         Age   Education
count  400.000000  400.000000  400.000000
mean   115.795000   53.322500   13.900000
std     23.676664   16.200297    2.620528
min     24.000000   25.000000   10.000000
25%    100.000000   39.750000   12.000000
50%    117.000000   54.500000   14.000000
75%    131.000000   66.000000   16.000000
max    191.000000   80.000000   18.000000

ShelveLoc
Medium    219
Bad        96
Good       85
Name: count, dtype: int64

Urban
Yes    282
No     118
Name: count, dtype: int64

US
Yes    258
No     142
Name: count, dtype: int64
```

In [12]:
```
X = pd.get_dummies(df2, columns=["ShelveLoc", "Urban", "US"])
for column in X.select_dtypes("bool"):
    X[column] = X[column].astype(int)

X = X.drop(columns=["Sales"])
```

```
X = sm.add_constant(X)
Y = df2["Sales"]
model = sm.OLS(Y, X).fit()

print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.873
Model:                            OLS   Adj. R-squared:                  0.870
Method:                 Least Squares   F-statistic:                     243.4
Date:                Mon, 09 Feb 2026   Prob (F-statistic):          1.60e-166
Time:                        06:38:55   Log-Likelihood:                -568.99
No. Observations:                 400   AIC:                             1162.
Df Residuals:                     388   BIC:                             1210.
Df Model:                          11
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const                3.3853      0.253     13.370      0.000       2.887       3.883
CompPrice            0.0928      0.004     22.378      0.000       0.085       0.101
Income               0.0158      0.002      8.565      0.000       0.012       0.019
Advertising          0.1231      0.011     11.066      0.000       0.101       0.145
Population           0.0002      0.000      0.561      0.575      -0.001       0.001
Price               -0.0954      0.003    -35.700      0.000      -0.101      -0.090
Age                 -0.0460      0.003    -14.472      0.000      -0.052      -0.040
Education           -0.0211      0.020     -1.070      0.285      -0.060       0.018
ShelveLoc_Bad       -1.1405      0.118     -9.629      0.000      -1.373      -0.908
ShelveLoc_Good       3.7096      0.121     30.652      0.000       3.472       3.948
ShelveLoc_Medium     0.8162      0.107      7.605      0.000       0.605       1.027
Urban_No             1.6312      0.138     11.789      0.000       1.359       1.903
Urban_Yes            1.7541      0.139     12.629      0.000       1.481       2.027
US_No                1.7847      0.146     12.243      0.000       1.498       2.071
US_Yes               1.6006      0.148     10.783      0.000       1.309       1.892
==============================================================================
Omnibus:                        0.811   Durbin-Watson:                   2.013
Prob(Omnibus):                  0.667   Jarque-Bera (JB):                0.765
Skew:                           0.107   Prob(JB):                        0.682
Kurtosis:                       2.994   Cond. No.                     3.32e+18
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The smallest eigenvalue is 4.43e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Conclusion from model summary

The model achieves an R-squared value of 0.873, which indicates that approximately 87.3% of the variance in sales can be explained by the model. The prob F-statistic (1.60e-166) is much less than 0.05, indicating that the model is statistically significant overall.

Coefficient Significance: When interpreting individual coefficients, we should examine their confidence intervals. Coefficients whose 95% confidence intervals do not include zero are statistically significant predictors. The width of the confidence interval indicates the precision of our estimate - narrower intervals suggest more precise estimates.

In [13]:
```
X = df2.drop(columns=["Sales", "Population", "Education", "Age", "Urban", "US"])
X = pd.get_dummies(X, columns=["ShelveLoc"])

for column in X.select_dtypes("bool"):
    X[column] = X[column].astype(int)

X["Income:Advertising"] = df2["Income"] * df2["Advertising"]
X["Price:Age"] = df2["Price"] * df2["Age"]
Y = df2["Sales"]
X = sm.add_constant(X)
model = sm.OLS(Y, X).fit()

print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.870
Model:                            OLS   Adj. R-squared:                  0.868
Method:                 Least Squares   F-statistic:                     328.2
Date:                Mon, 09 Feb 2026   Prob (F-statistic):          2.90e-168
Time:                        06:38:55   Log-Likelihood:                -573.74
No. Observations:                 400   AIC:                             1165.
Df Residuals:                     391   BIC:                             1201.
Df Model:                           8
Covariance Type:            nonrobust
====================================================================================
==
                         coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------------
--
const                  4.1957      0.352     11.903      0.000       3.503       4.8
89
CompPrice              0.0934      0.004     22.492      0.000       0.085       0.1
02
Income                 0.0098      0.003      3.756      0.000       0.005       0.0
15
Advertising            0.0534      0.021      2.544      0.011       0.012       0.0
95
Price                 -0.0759      0.003    -25.591      0.000      -0.082      -0.0
70
ShelveLoc_Bad         -0.8966      0.143     -6.280      0.000      -1.177      -0.6
16
ShelveLoc_Good         3.9982      0.149     26.769      0.000       3.705       4.2
92
ShelveLoc_Medium       1.0942      0.133      8.221      0.000       0.833       1.3
56
Income:Advertising     0.0009      0.000      3.124      0.002       0.000       0.0
01
Price:Age             -0.0004   2.69e-05    -13.713      0.000      -0.000      -0.0
00
==============================================================================
Omnibus:                        1.537   Durbin-Watson:                   1.988
Prob(Omnibus):                  0.464   Jarque-Bera (JB):                1.326
Skew:                           0.129   Prob(JB):                        0.515
Kurtosis:                       3.116   Cond. No.                     2.80e+19
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The smallest eigenvalue is 2.18e-29. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Conclusion from model summary

This model achieves an R-squared value of 0.870, which is slightly worse than the previous model (0.873). The prob F-statistic (2.90e-168) indicates that the model is statistically significant overall ($p < 0.05$).

Coefficient Interpretation: When examining individual coefficients, we should check their confidence intervals to assess both significance (whether the interval includes zero) and precision (width of the interval). Some predictors may become insignificant when used together with other predictors due to multicollinearity - when predictor variables are highly correlated with each other. In such cases, a predictor that appears significant in isolation may become insignificant in a multiple regression model because its information is already captured by correlated predictors.

Beat the teacher

In [14]:
```
X = df2.drop(columns=["Sales"])
X = pd.get_dummies(X, columns=["ShelveLoc", "US", "Urban"])

for column in X.select_dtypes("bool"):
    X[column] = X[column].astype(int)

X["Income:Advertising"] = df2["Income"] * df2["Advertising"]
Y = df2["Sales"]
X = sm.add_constant(X)
model = sm.OLS(Y, X).fit()

print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.876
Model:                            OLS   Adj. R-squared:                  0.872
Method:                 Least Squares   F-statistic:                     227.6
Date:                Mon, 09 Feb 2026   Prob (F-statistic):          5.48e-167
Time:                        06:38:55   Log-Likelihood:                -565.00
No. Observations:                 400   AIC:                             1156.
Df Residuals:                     387   BIC:                             1208.
Df Model:                          12
Covariance Type:            nonrobust
==============================================================================
==
                         coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
--
const                  3.5106      0.255     13.767      0.000       3.009       4.0
12
CompPrice              0.0931      0.004     22.630      0.000       0.085       0.1
01
Income                 0.0107      0.003      4.125      0.000       0.006       0.0
16
Advertising            0.0684      0.022      3.043      0.003       0.024       0.1
13
Population             0.0002      0.000      0.456      0.649      -0.001       0.0
01
Price                 -0.0952      0.003    -35.962      0.000      -0.100      -0.0
90
Age                   -0.0454      0.003    -14.367      0.000      -0.052      -0.0
39
Education             -0.0220      0.020     -1.125      0.261      -0.060       0.0
16
ShelveLoc_Bad         -1.1051      0.118     -9.356      0.000      -1.337      -0.8
73
ShelveLoc_Good         3.7570      0.121     31.005      0.000       3.519       3.9
95
ShelveLoc_Medium       0.8586      0.107      7.988      0.000       0.647       1.0
70
US_No                  1.8361      0.146     12.604      0.000       1.550       2.1
23
US_Yes                 1.6744      0.150     11.199      0.000       1.380       1.9
68
Urban_No               1.6884      0.139     12.173      0.000       1.416       1.9
61
Urban_Yes              1.8222      0.140     13.031      0.000       1.547       2.0
97
Income:Advertising     0.0008      0.000      2.791      0.006       0.000       0.0
01
==============================================================================
Omnibus:                        1.390   Durbin-Watson:                   2.036
Prob(Omnibus):                  0.499   Jarque-Bera (JB):                1.229
Skew:                           0.131   Prob(JB):                        0.541
Kurtosis:                       3.070   Cond. No.                     1.15e+19
==============================================================================
```

```
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The smallest eigenvalue is 1.73e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Conclusion from Beat the teacher model

This model achieves an R-squared value of 0.876, which is slightly better than the previous models. The prob F-statistic (5.48e-167) indicates high statistical significance ($p < 0.05$). By including all available predictors and adding an interaction term between `Income` and `Advertising`, the model captures the synergistic effect where the impact of income on sales may depend on advertising levels (or vice versa).

Model Selection Considerations: When comparing models, we should consider:

1. **Confidence Intervals**: Check that key predictors have confidence intervals that exclude zero (indicating significance) and assess the precision of estimates.
2. **Multicollinearity**: If predictors are highly correlated, some may appear insignificant even if they contain useful information.
3. **Practical Significance**: Beyond statistical significance, consider whether the improvement in R-squared justifies the added model complexity.