Multiple Linear Regression in StatsModels - Lab

Introduction

In this lab, you'll practice fitting a multiple linear regression model on the Ames Housing dataset!

Objectives

You will be able to:

- Perform a multiple linear regression using StatsModels
- Visualize individual predictors within a multiple linear regression
- Interpret multiple linear regression coefficients from raw, un-transformed data

The Ames Housing Dataset

The Ames Housing dataset is a newer (2011) replacement for the classic Boston Housing dataset. Each record represents a residential property sale in Ames, Iowa. It contains many different potential predictors and the target variable is `SalePrice`.

```
import pandas as pd
ames = pd.read_csv("ames.csv", index_col=0)
ames
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	Lo
Id							
1	60	RL	65.0	8450	Pave	NaN	Re
2	20	RL	80.0	9600	Pave	NaN	Re
3	60	RL	68.0	11250	Pave	NaN	IR1
4	70	RL	60.0	9550	Pave	NaN	IR1
5	60	RL	84.0	14260	Pave	NaN	IR1
...	...	...	...	...	...	...	...
1456	60	RL	62.0	7917	Pave	NaN	Re
1457	20	RL	85.0	13175	Pave	NaN	Re
1458	70	RL	66.0	9042	Pave	NaN	Re
1459	20	RL	68.0	9717	Pave	NaN	Re

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	Lo
Id							
1460	20	RL	75.0	9937	Pave	NaN	Re

1460 rows × 80 columns

```
ames.describe()
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCo
count	1460.000000	1201.000000	1460.000000	1460.000000	1460.0000
mean	56.897260	70.049958	10516.828082	6.099315	5.575342
std	42.300571	24.284752	9981.264932	1.382997	1.112799
min	20.000000	21.000000	1300.000000	1.000000	1.000000
25%	20.000000	59.000000	7553.500000	5.000000	5.000000
50%	50.000000	69.000000	9478.500000	6.000000	5.000000
75%	70.000000	80.000000	11601.500000	7.000000	6.000000
max	190.000000	313.000000	215245.000000	10.000000	9.000000

8 rows × 37 columns

We will focus specifically on a subset of the overall dataset. These features are:

```
LotArea: Lot size in square feet
```

```
1stFlrSF: First Floor square feet

GrLivArea: Above grade (ground) living area square feet
```

```python
ames_subset = ames[['LotArea', '1stFlrSF', 'GrLivArea', 'SalePrice']].copy()
ames_subset
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

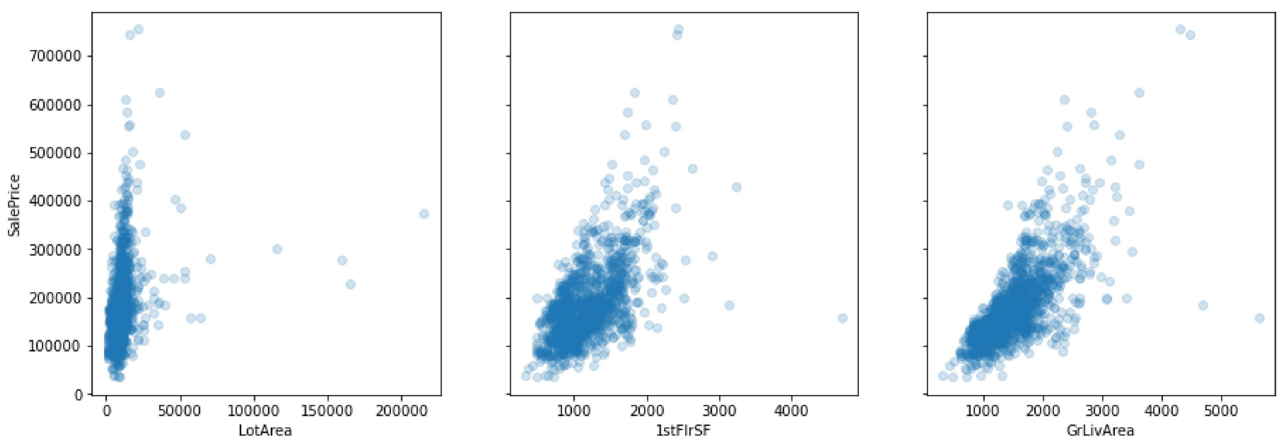
	LotArea	1stFlrSF	GrLivArea	SalePrice
Id				
1	8450	856	1710	208500
2	9600	1262	1262	181500
3	11250	920	1786	223500
4	9550	961	1717	140000
5	14260	1145	2198	250000
...	...	...	...	...
1456	7917	953	1647	175000
1457	13175	2073	2073	210000
1458	9042	1188	2340	266500
1459	9717	1078	1078	142125
1460	9937	1256	1256	147500

1460 rows × 4 columns

Step 1: Visualize Relationships Between Features and Target

For each feature in the subset, create a scatter plot that shows the feature on the x-axis and `SalePrice` on the y-axis.

```
import matplotlib.pyplot as plt

fig, axes = plt.subplots(ncols=3, figsize=(15,5), sharey=True)
axes[0].set_ylabel("SalePrice")

for i, col in enumerate(ames_subset.drop("SalePrice", axis=1).columns):
    ax = axes[i]
    ax.scatter(ames_subset[col], ames_subset["SalePrice"], alpha=0.2)
    ax.set_xlabel(col)
```




```
"""
All three of these features seem to have a linear relationship with SalePrice

1stFlrSF seems to have the most variance vs. SalePrice

All three have a few outliers that could potentially skew the results
"""
```

Step 2: Build a Simple Linear Regression Model

Set the dependent variable (`y`) to be the `SalePrice`, then choose one of the features shown in the subset above to be the baseline independent variable (`X`).

Build a linear regression using StatsModels, describe the overall model performance, and interpret its coefficients.

```
# Explore correlation to find a good starting point
ames_subset.corr()["SalePrice"]
```

```
LotArea      0.263843
1stFlrSF     0.605852
GrLivArea    0.708624
SalePrice    1.000000
Name: SalePrice, dtype: float64
```

```
y = ames_subset["SalePrice"]
# Above grade living area had the highest correlation
X_baseline = ames_subset[["GrLivArea"]]
```

```
import statsmodels.api as sm

baseline_model = sm.OLS(y, sm.add_constant(X_baseline))
baseline_results = baseline_model.fit()

print(baseline_results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              SalePrice   R-squared:                       0.502
Model:                            OLS   Adj. R-squared:                  0.502
Method:                 Least Squares   F-statistic:                     1471.
Date:                Mon, 09 May 2022   Prob (F-statistic):          4.52e-223
Time:                        19:15:03   Log-Likelihood:                -18035.
No. Observations:                1460   AIC:                         3.607e+04
Df Residuals:                    1458   BIC:                         3.608e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1.857e+04   4480.755      4.144      0.000    9779.612    2.74e+04
GrLivArea    107.1304      2.794     38.348      0.000     101.650     112.610
==============================================================================
Omnibus:                      261.166   Durbin-Watson:                   2.025
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3432.287
Skew:                           0.410   Prob(JB):                         0.00
Kurtosis:                      10.467   Cond. No.                     4.90e+03
==============================================================================
```

```
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 4.9e+03. This might indicate that there are
strong multicollinearity or other numerical problems.



"""
Our model is statistically significant overall, and explains about 50% of the varian
in SalePrice.

Both our intercept and our coefficient for GrLivArea are statistically significant.

Our intercept is about 18,600, meaning that a home with 0 square feet of above-groun
living area would cost about $18.6k.

Our coefficient for GrLivArea is about 107, which means that for each additional squ
foot of above ground living area, we expect the price to increase about $107.
"""
```

Step 3: Build a Multiple Linear Regression Model

For this model, use **all of** the features in `ames_subset`.

```
X = ames_subset.drop("SalePrice", axis=1)



subset_model = sm.OLS(y, sm.add_constant(X))
subset_results = subset_model.fit()

print(subset_results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              SalePrice   R-squared:                       0.565
Model:                            OLS   Adj. R-squared:                  0.564
Method:                 Least Squares   F-statistic:                     630.3
Date:                Mon, 09 May 2022   Prob (F-statistic):         1.57e-262
Time:                        19:15:09   Log-Likelihood:                -17936.
No. Observations:                1460   AIC:                         3.588e+04
Df Residuals:                    1456   BIC:                         3.590e+04
Df Model:                           3
Covariance Type:            nonrobust
```

```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -1.431e+04   4776.331     -2.997      0.003   -2.37e+04   -4944.183
LotArea        0.2841      0.145      1.956      0.051      -0.001       0.569
1stFlrSF      60.2866      4.388     13.739      0.000      51.679      68.894
GrLivArea     80.6061      3.193     25.248      0.000      74.344      86.869
==============================================================================
Omnibus:                      399.604   Durbin-Watson:                   1.996
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            13445.161
Skew:                          -0.588   Prob(JB):                         0.00
Kurtosis:                      17.820   Cond. No.                     5.07e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 5.07e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

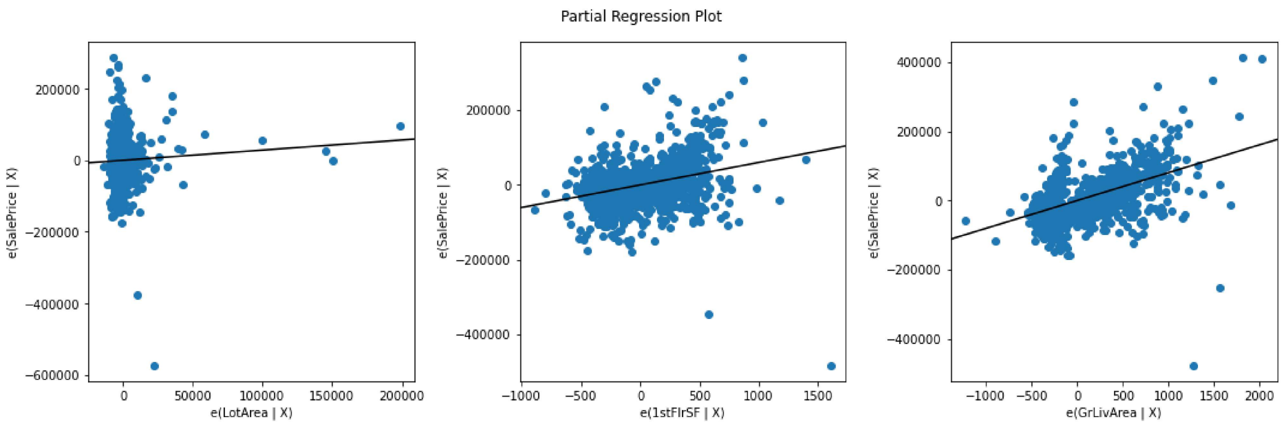
```
"""
Our new model is statistically significant overall, and explains about 57% of the
variance in SalePrice. This is about 7% more variance explained than the simple
model.

Using an alpha of 0.05, our intercept and coefficients for 1stFlrSF and GrLivArea
are statistically significant, but not our coefficient for LotArea.

Both our intercept and our coefficient for GrLivArea are statistically significant.

So, we have an improvement in terms of variance explained (R-Squared), but also
some values are not statistically significant. It depends on the use case whether
this model would be considered "better".
"""
```

Step 4: Create Partial Regression Plots for Features

Using your model from Step 3, visualize each of the features using partial regression plots.

```
fig = plt.figure(figsize=(15,5))
sm.graphics.plot_partregress_grid(
    subset_results,
    exog_idx=list(X.columns),
    grid=(1,3),
```

```
    fig=fig)
plt.tight_layout()
plt.show()
```




```
"""
In the context of a multiple regression model, LotArea seems to be a much weaker
predictor than it initially seemed. The partial regression plot is showing only the
variance in SalePrice that is not already explained by the other variables

1stFlrSF and GrLivArea look roughly the same as they did as standalone scatter
plots, although the slopes are not as steep.

Thinking back to the meaning of these variables, you might have guessed that
1stFlrSF and GrLivArea would have more overlap in the variance they explain, since
they are both related to the square footage of the house. However it seems that
they actually contain different enough information.

You also might notice that the outliers in LotArea might be having more of an impact
than anticipated. That best-fit line might not be where you intuitively would have
drawn it.
"""
```

Level Up (Optional)

Re-create this model in scikit-learn, and check if you get the same R-Squared and coefficients.

```
from sklearn.linear_model import LinearRegression

lr = LinearRegression()
lr.fit(X, y)
```

```
LinearRegression()
```

```
subset_results.rsquared
```

```
0.5649801771384368
```

```
lr.score(X, y)
```

```
0.5649801771384368
```

```
subset_results.params.values
```

```
array([-1.43134089e+04,  2.84133589e-01,  6.02866463e+01,  8.06060583e+01])
```

```
import numpy as np
np.append(lr.intercept_, lr.coef_)
```

```
array([-1.43134089e+04,  2.84133589e-01,  6.02866463e+01,  8.06060583e+01])
```

Summary

Congratulations! You fitted your first multiple linear regression model on the Ames Housing data using StatsModels.

Releases

No releases published

Packages

No packages published

Contributors 4

- **LoreDirick** Lore Dirick
- **hoffm386** Erin R Hoffman
- **mas16** matt
- **sumedh10** Sumedh Panchadhar

Languages

- **Jupyter Notebook** 100.0%