learn-co-curriculum / dsc-interaction-terms-lab-v2-5 Public

View license

0 stars 4 forks

Star Watch

Code Issues Pull requests Actions Projects Security Insights

solution

This branch is 4 commits ahead, 7 commits behind master.

hoffm386 inferential framing, more interpretation of plots ... on Jun 10 8

View code

README.md

Interactions - Lab

Introduction

In this lab, you'll explore interactions in the Ames Housing dataset.

Objectives

You will be able to:

- Determine if an interaction term would be useful for a specific model or set of data
- Create interaction terms out of independent variables in linear regression
- Interpret coefficients of linear regression models that contain interaction terms

Ames Housing Data

Once again we will be using the Ames Housing dataset, where each record represents a home sale:

```
import pandas as pd

ames = pd.read_csv('ames.csv', index_col=0)

# Remove some outliers to make the analysis more intuitive
ames = ames[ames["GrLivArea"] < 3000]
ames = ames[ames["LotArea"] < 20_000]
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	IR
4	70	RL	60.0	9550	Pave	NaN	IR
5	60	RL	84.0	14260	Pave	NaN	IR
...	...	...	...	...	...	...	...
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

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	Lo
Id							
1459	20	RL	68.0	9717	Pave	NaN	Re
1460	20	RL	75.0	9937	Pave	NaN	Re

1396 rows × 80 columns

In particular, we'll use these numeric and categorical features:

```
numeric = ['LotArea', '1stFlrSF', 'GrLivArea']
categorical = ['KitchenQual', 'Neighborhood']
```

Build a Baseline Model

Initial Data Preparation

Use all of the numeric and categorical features described above. (We will call this the "baseline" model because we are making a comparison with and without an interaction term. In a complete modeling process you would start with a simpler baseline.)

One-hot encode the categorical features (dropping the first), and center (subtract the mean) from the numeric features.

```
# Select relevant numeric features and center them
ames_numeric = ames[numeric].copy()
for column in numeric:
    ames_numeric[column] = ames_numeric[column] - ames_numeric[column].mean()
ames_numeric
```

<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
Id			
1	-865.191977	-284.866046	231.095272
2	284.808023	121.133954	-216.904728
3	1934.808023	-220.866046	307.095272
4	234.808023	-179.866046	238.095272
5	4944.808023	4.133954	719.095272
...	...	...	...
1456	-1398.191977	-187.866046	168.095272
1457	3859.808023	932.133954	594.095272
1458	-273.191977	47.133954	861.095272
1459	401.808023	-62.866046	-400.904728
1460	621.808023	115.133954	-222.904728

1396 rows × 3 columns

```
# Select relevant categorical features and one-hot encode them
ames_categorical = ames[categorical].copy()
ames_categorical = pd.get_dummies(ames_categorical, columns=categorical, drop_first=
ames_categorical
```

<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>

	KitchenQual_Fa	KitchenQual_Gd	KitchenQual_TA	Neighborhood_Blue
Id				
1	0	1	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	0	0
5	0	1	0	0
...	...	...	...	...
1456	0	0	1	0
1457	0	0	1	0
1458	0	1	0	0
1459	0	1	0	0
1460	0	0	1	0

1396 rows × 27 columns

Build a Model with No Interaction Terms

Using the numeric and categorical features that you have prepared, as well as `SalePrice` as the target, build a StatsModels OLS model.

```
import statsmodels.api as sm

y = ames["SalePrice"]
X_baseline = pd.concat([ames_numeric, ames_categorical], axis=1)

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

Evaluate the Model without Interaction Terms

Describe the adjusted R-Squared as well as which coefficients are statistically significant. For now you can skip interpreting all of the coefficients.

```
print(baseline_results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              SalePrice   R-squared:                       0.831
Model:                            OLS   Adj. R-squared:                  0.827
Method:                 Least Squares   F-statistic:                     223.6
Date:                Fri, 10 Jun 2022   Prob (F-statistic):               0.00
Time:                        16:06:55   Log-Likelihood:                -16370.
No. Observations:                1396   AIC:                         3.280e+04
Df Residuals:                    1365   BIC:                         3.297e+04
Df Model:                          30
Covariance Type:            nonrobust
=================================================================================
                           coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-------
const                 2.544e+05   8514.941     29.874      0.000    2.38e+05
2.71e+05
LotArea                  2.6298      0.333      7.889      0.000       1.976
3.284
1stFlrSF                33.6365      3.110     10.816      0.000      27.536
39.737
GrLivArea               50.9761      2.423     21.043      0.000      46.224
55.728
KitchenQual_Fa       -8.968e+04   6630.761    -13.525      0.000   -1.03e+05
-7.67e+04
KitchenQual_Gd       -5.419e+04   3894.115    -13.916      0.000   -6.18e+04
-4.66e+04
KitchenQual_TA       -7.457e+04   4259.873    -17.505      0.000   -8.29e+04
-6.62e+04
Neighborhood_Blueste  -778.8650   2.29e+04     -0.034      0.973   -4.56e+04
4.41e+04
Neighborhood_BrDale  -2.098e+04    1.1e+04     -1.914      0.056   -4.25e+04
526.487
Neighborhood_BrkSide -2.962e+04   8842.840     -3.350      0.001    -4.7e+04
-1.23e+04
Neighborhood_ClearCr -1.335e+04   1.13e+04     -1.180      0.238   -3.56e+04
8849.979
Neighborhood_CollgCr -2624.4150   8113.003     -0.323      0.746   -1.85e+04
1.33e+04
Neighborhood_Crawfor -3265.4285   9091.338     -0.359      0.720   -2.11e+04
1.46e+04
Neighborhood_Edwards -4.239e+04   8498.055     -4.988      0.000   -5.91e+04
-2.57e+04
Neighborhood_Gilbert -4720.8057   8737.465     -0.540      0.589   -2.19e+04
```

```
1.24e+04
Neighborhood_IDOTRR  -4.937e+04   9419.188     -5.242      0.000   -6.78e+04
-3.09e+04
Neighborhood_MeadowV -3.301e+04   1.07e+04     -3.081      0.002    -5.4e+04
-1.2e+04
Neighborhood_Mitchel -1.746e+04   9138.562     -1.910      0.056   -3.54e+04
469.180
Neighborhood_NAmes   -3.292e+04   8170.560     -4.029      0.000   -4.89e+04
-1.69e+04
Neighborhood_NPkVill -4518.3147   1.27e+04     -0.356      0.722   -2.94e+04
2.04e+04
Neighborhood_NWAmes  -2.478e+04   8738.425     -2.836      0.005   -4.19e+04
-7636.773
Neighborhood_NoRidge  3.584e+04   9541.904      3.756      0.000    1.71e+04
5.46e+04
Neighborhood_NridgHt  4.642e+04   8561.756      5.421      0.000    2.96e+04
6.32e+04
Neighborhood_OldTown -5.093e+04   8358.005     -6.094      0.000   -6.73e+04
-3.45e+04
Neighborhood_SWISU   -4.662e+04   1.01e+04     -4.624      0.000   -6.64e+04
-2.68e+04
Neighborhood_Sawyer  -3.282e+04   8751.214     -3.751      0.000      -5e+04
-1.57e+04
Neighborhood_SawyerW -1.716e+04   8718.180     -1.968      0.049   -3.43e+04
-57.735
Neighborhood_Somerst  1.728e+04   8275.364      2.088      0.037    1045.592
3.35e+04
Neighborhood_StoneBr  5.296e+04   9791.726      5.409      0.000    3.38e+04
7.22e+04
Neighborhood_Timber   5378.3716   9474.818      0.568      0.570   -1.32e+04
2.4e+04
Neighborhood_Veenker  4649.5842   1.24e+04      0.374      0.709   -1.98e+04
2.91e+04
==============================================================================
Omnibus:                      271.896   Durbin-Watson:                   2.042
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1864.321
Skew:                           0.721   Prob(JB):                         0.00
Kurtosis:                       8.475   Cond. No.                     1.61e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.61e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
# Which was the reference neighborhood? This will be the first one alphabetically
ames["Neighborhood"].sort_values().unique()
```

```
array(['Blmngtn', 'Blueste', 'BrDale', 'BrkSide', 'ClearCr', 'CollgCr',
       'Crawfor', 'Edwards', 'Gilbert', 'IDOTRR', 'MeadowV', 'Mitchel',
       'NAmes', 'NPkVill', 'NWAmes', 'NoRidge', 'NridgHt', 'OldTown',
       'SWISU', 'Sawyer', 'SawyerW', 'Somerst', 'StoneBr', 'Timber',
       'Veenker'], dtype=object)
```

▶ Answer (click to reveal)

Identify Good Candidates for Interaction Terms

Numeric x Categorical Term

Square footage of a home is often worth different amounts depending on the neighborhood. So let's see if we can improve the model by building an interaction term between `GrLivArea` and one of the `Neighborhood` categories.

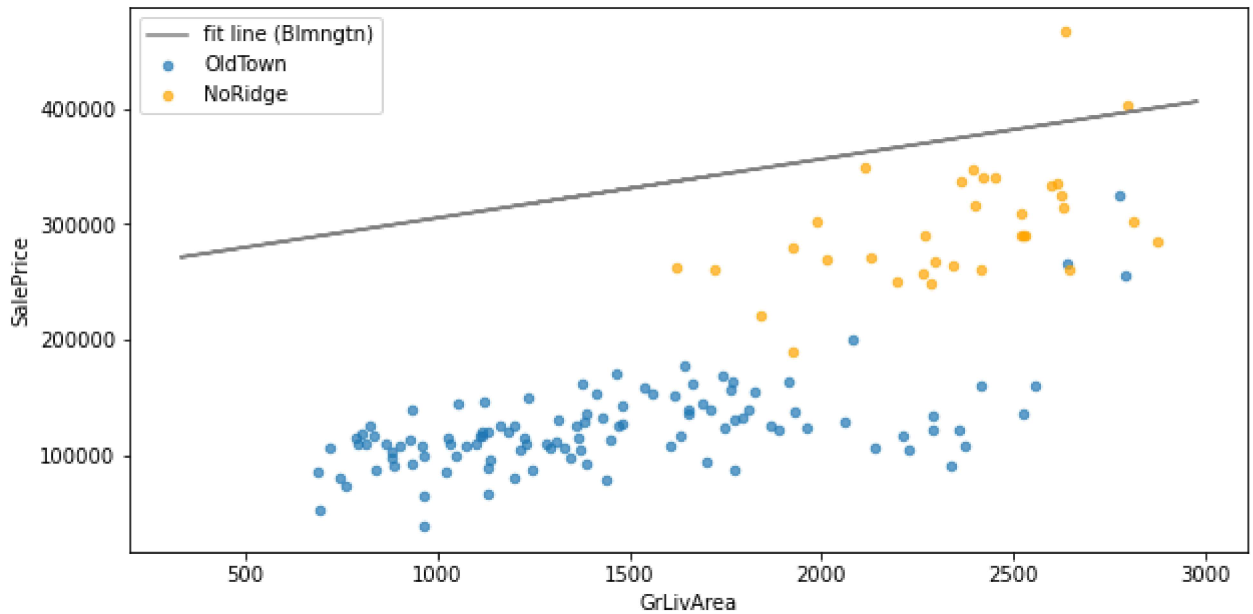
Because there are so many neighborhoods to consider, we'll narrow it down to 2 options: `Neighborhood_OldTown` or `Neighborhood_NoRidge` .

First, create a plot that has:

- `GrLivArea` on the x-axis
- `SalePrice` on the y-axis
- A scatter plot of homes in the `OldTown` and `NoRidge` neighborhoods, identified by color
    - Hint: you will want to call `.scatter` twice, once for each neighborhood
- A line showing the fit of `GrLivArea` vs. `SalePrice` for the reference neighborhood

```
import matplotlib.pyplot as plt

# Filter to houses in specific neighborhoods
oldtown = ames[ames["Neighborhood"] == "OldTown"]
noridge = ames[ames["Neighborhood"] == "NoRidge"]

fig, ax = plt.subplots(figsize=(10,5))

# Create scatter plots with 2 different colors
oldtown.plot.scatter(x="GrLivArea", y="SalePrice", alpha=0.7, label="OldTown", ax=ax
```

```
noridge.plot.scatter(x="GrLivArea", y="SalePrice", alpha=0.7, color="orange", label=

# Plot best fit line
intercept = baseline_results.params["const"]
slope = baseline_results.params["GrLivArea"]
ax.plot(ames["GrLivArea"], intercept + ames["GrLivArea"] * slope, color="gray", labe

ax.legend();
```



Looking at this plot, do either of these neighborhoods seem to have a **slope** that differs notably from the best fit line? If so, this is an indicator that an interaction term might be useful.

Identify what, if any, interaction terms you would create based on this information.

▶ **Answer (click to reveal)**

Numeric x Numeric Term

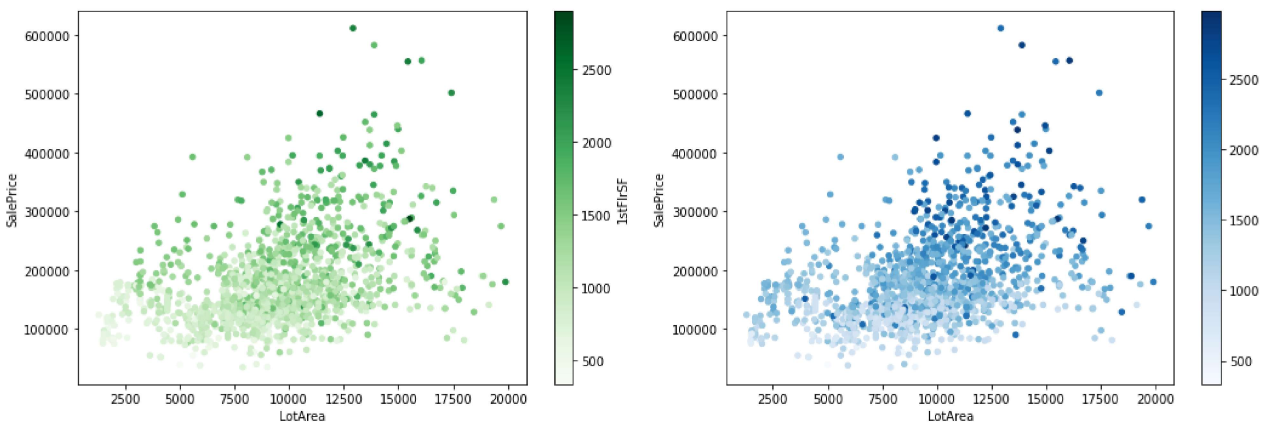
Let's also investigate to see whether adding an interaction term between two of the numeric features would be helpful.

We'll specifically focus on interactions with `LotArea`. Does the value of an extra square foot of lot area change depending on the square footage of the home? Both `1stFlrSF` and `GrLivArea` are related to home square footage, so we'll use those in our comparisons.

Create two side-by-side plots:

1. One scatter plot of `LotArea` vs. `SalePrice` where the color of the points is based on `1stFlrSF`
2. One scatter plot of `LotArea` vs. `SalePrice` where the color of the points is based on `GrLivArea`

```
fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(15,5))
ames.plot.scatter(x="LotArea", y="SalePrice", c="1stFlrSF", cmap="Greens", ax=ax1)
ames.plot.scatter(x="LotArea", y="SalePrice", c="GrLivArea", cmap="Blues", ax=ax2)
fig.tight_layout();
```



Looking at these plots, does the slope between `LotArea` and `SalePrice` seem to differ based on the color of the point? If it does, that is an indicator that an interaction term might be helpful.

Describe your interpretation below:

▶ **Answer (click to reveal)**

Build and Interpret a Model with Interactions

Build a Second Model

Based on your analysis above, build a model based on the baseline model with one or more interaction terms added.

```
X_interaction = pd.concat([ames_numeric, ames_categorical], axis=1)

X_interaction["GrLivArea x Neighborhood_NoRidge"] = X_interaction["GrLivArea"] * \
                        X_interaction["Neighborhood_NoRidge"]
X_interaction["LotArea x 1stFlrSF"] = X_interaction["LotArea"] * X_interaction["1stF
```

```
interaction_model = sm.OLS(y, sm.add_constant(X_interaction))
interaction_results = interaction_model.fit()
```

Evaluate the Model with Interactions

Same as with the baseline model, describe the adjusted R-Squared and statistical significance of the coefficients.

```
print(interaction_results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              SalePrice   R-squared:                       0.833
Model:                            OLS   Adj. R-squared:                  0.829
Method:                 Least Squares   F-statistic:                     212.1
Date:                Fri, 10 Jun 2022   Prob (F-statistic):               0.00
Time:                        16:07:17   Log-Likelihood:                -16363.
No. Observations:                1396   AIC:                         3.279e+04
Df Residuals:                    1363   BIC:                         3.296e+04
Df Model:                          32
Covariance Type:            nonrobust
========================================================================================
                                       coef    std err          t      P>|t|
[0.025      0.975]
----------------------------------------------------------------------------------
-------------------
const                             2.584e+05   8543.049     30.244      0.000
2.42e+05    2.75e+05
LotArea                              2.5810      0.333      7.756      0.000
1.928       3.234
1stFlrSF                            30.5397      3.206      9.526      0.000
24.251      36.829
GrLivArea                           50.9848      2.432     20.964      0.000
46.214      55.756
KitchenQual_Fa                   -8.869e+04   6605.188    -13.428      0.000
-1.02e+05   -7.57e+04
KitchenQual_Gd                   -5.295e+04   3890.488    -13.609      0.000
-6.06e+04   -4.53e+04
KitchenQual_TA                   -7.315e+04   4257.029    -17.182      0.000
-8.15e+04   -6.48e+04
Neighborhood_Blueste             -1.865e+04    2.32e+04     -0.802      0.423
-6.42e+04     2.7e+04
Neighborhood_BrDale              -3.982e+04     1.2e+04     -3.319      0.001
-6.34e+04   -1.63e+04
Neighborhood_BrkSide             -3.752e+04   9042.493     -4.149      0.000
```

```
-5.53e+04   -1.98e+04
Neighborhood_ClearCr             -1.802e+04   1.13e+04     -1.591      0.112
-4.03e+04    4206.704
Neighborhood_CollgCr             -8354.3245   8214.121     -1.017      0.309
-2.45e+04    7759.366
Neighborhood_Crawfor             -9735.5393   9208.642     -1.057      0.291
-2.78e+04    8329.109
Neighborhood_Edwards             -4.874e+04   8620.307     -5.654      0.000
-6.57e+04   -3.18e+04
Neighborhood_Gilbert             -1.054e+04   8830.991     -1.193      0.233
-2.79e+04    6785.869
Neighborhood_IDOTRR              -5.689e+04   9579.755     -5.938      0.000
-7.57e+04   -3.81e+04
Neighborhood_MeadowV             -4.831e+04   1.14e+04     -4.236      0.000
-7.07e+04   -2.59e+04
Neighborhood_Mitchel              -2.32e+04   9218.870     -2.517      0.012
-4.13e+04   -5118.644
Neighborhood_NAmes               -3.904e+04   8288.528     -4.710      0.000
-5.53e+04   -2.28e+04
Neighborhood_NPkVill             -1.473e+04   1.29e+04     -1.140      0.255
-4.01e+04    1.06e+04
Neighborhood_NWAmes              -3.069e+04   8835.536     -3.473      0.001
-4.8e+04   -1.34e+04
Neighborhood_NoRidge               1.86e+04   1.75e+04      1.065      0.287
-1.57e+04    5.29e+04
Neighborhood_NridgHt              3.971e+04   8708.015      4.560      0.000
2.26e+04    5.68e+04
Neighborhood_OldTown             -5.819e+04   8536.608     -6.817      0.000
-7.49e+04   -4.14e+04
Neighborhood_SWISU               -5.331e+04   1.02e+04     -5.231      0.000
-7.33e+04   -3.33e+04
Neighborhood_Sawyer              -3.917e+04   8866.300     -4.417      0.000
-5.66e+04   -2.18e+04
Neighborhood_SawyerW             -2.276e+04   8801.507     -2.586      0.010
-4e+04   -5494.517
Neighborhood_Somerst              9776.4928   8473.381      1.154      0.249
-6845.790    2.64e+04
Neighborhood_StoneBr               4.77e+04   9847.157      4.844      0.000
2.84e+04     6.7e+04
Neighborhood_Timber              -1827.4479   9620.049     -0.190      0.849
-2.07e+04     1.7e+04
Neighborhood_Veenker             -1633.5130   1.25e+04     -0.131      0.896
-2.62e+04    2.29e+04
GrLivArea x Neighborhood_NoRidge    12.9706     16.980      0.764      0.445
-20.340      46.281
LotArea x 1stFlrSF                   0.0027      0.001      3.771      0.000
0.001       0.004
==============================================================================
Omnibus:                      253.719   Durbin-Watson:                   2.037
```

```
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1760.599
Skew:                           0.654   Prob(JB):                         0.00
Kurtosis:                       8.344   Cond. No.                     7.29e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 7.29e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Interpret the Model Results

Interpret the coefficients for the intercept as well as the interactions and all variables used in the interactions. Make sure you only interpret the coefficients that were statistically significant!

```
interaction_results.params["const"]
```

```
258372.23042374293
```

```
interaction_results.params["LotArea"]
```

```
2.5810221682172148
```

```
interaction_results.params["1stFlrSF"]
```

```
30.539688663554905
```

```
interaction_results.params["LotArea x 1stFlrSF"]
```

```
0.0027043505209154973
```

▶ Answer (click to reveal)

Summary

You should now understand how to include interaction effects in your model! As you can see, interactions that seem promising may or may not end up being statistically significant. This is why exploration and iteration are important!

Releases

No releases published

Packages

No packages published

Contributors 4


LoreDirick Lore Dirick


hoffm386 Erin R Hoffman


sumedh10 Sumedh Panchadhar


fpolchow Forest Polchow

Languages

- **Jupyter Notebook** 100.0%