

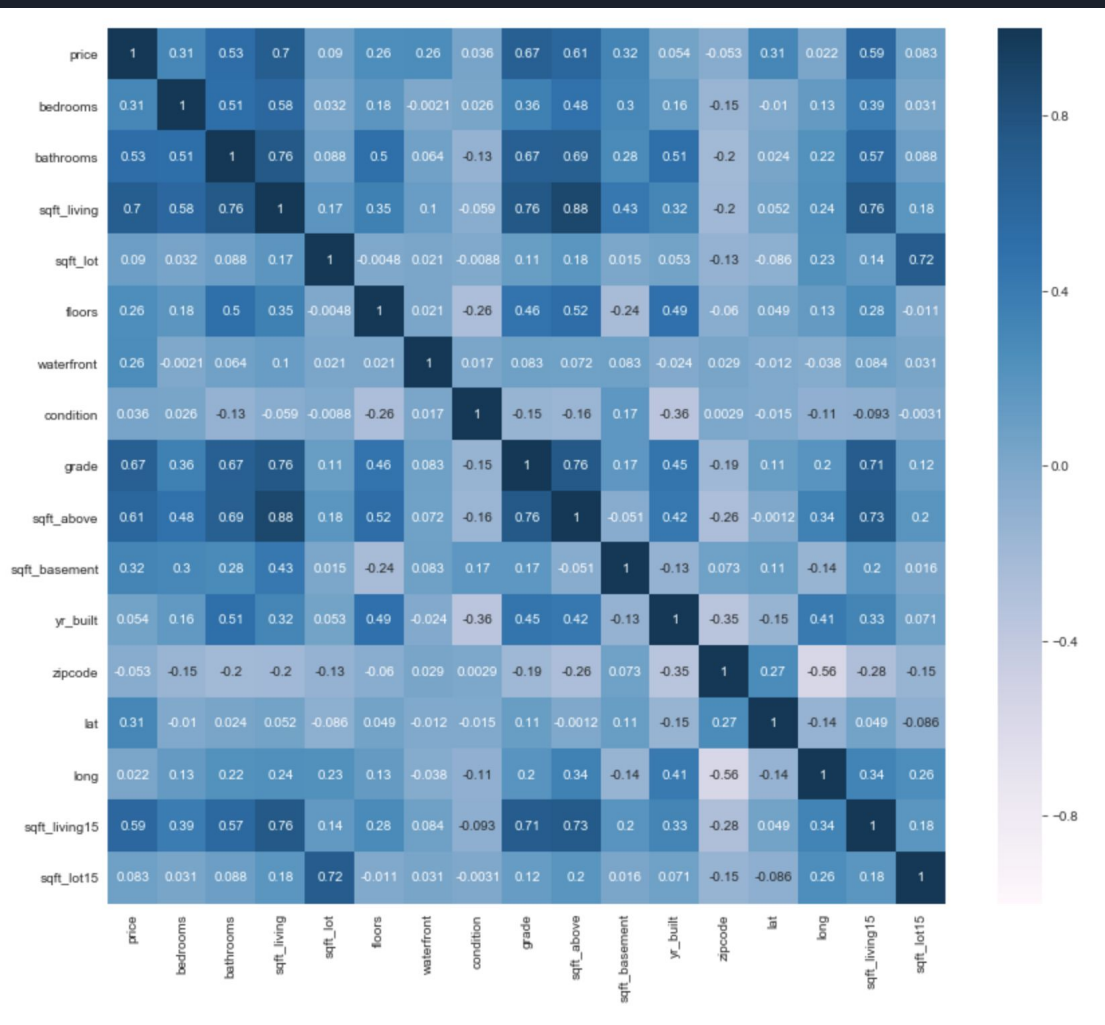


Predicting Home Values with Linear Regression



Contents of the Jupyter Notebook:

1. Methodology
2. Exploring the Data
3. Splitting, Selecting, and Scaling Data
4. Linear Regression
5. Takeaways



Creating dummy variables:

```
# Create dummy variables
from sklearn import preprocessing
from sklearn.preprocessing import KBinsDiscretizer

waterfront_dummies = pd.get_dummies(df['waterfront'], prefix='wf', drop_first=True)
condition_dummies = pd.get_dummies(df['condition'], prefix='cond', drop_first=True)
```

Binning continuous values:

```
disc_5k = KBinsDiscretizer(encode='onehot-dense', strategy='kmeans')
disc_5 = KBinsDiscretizer(encode='onehot-dense')
disc_3 = KBinsDiscretizer(encode='onehot-dense')

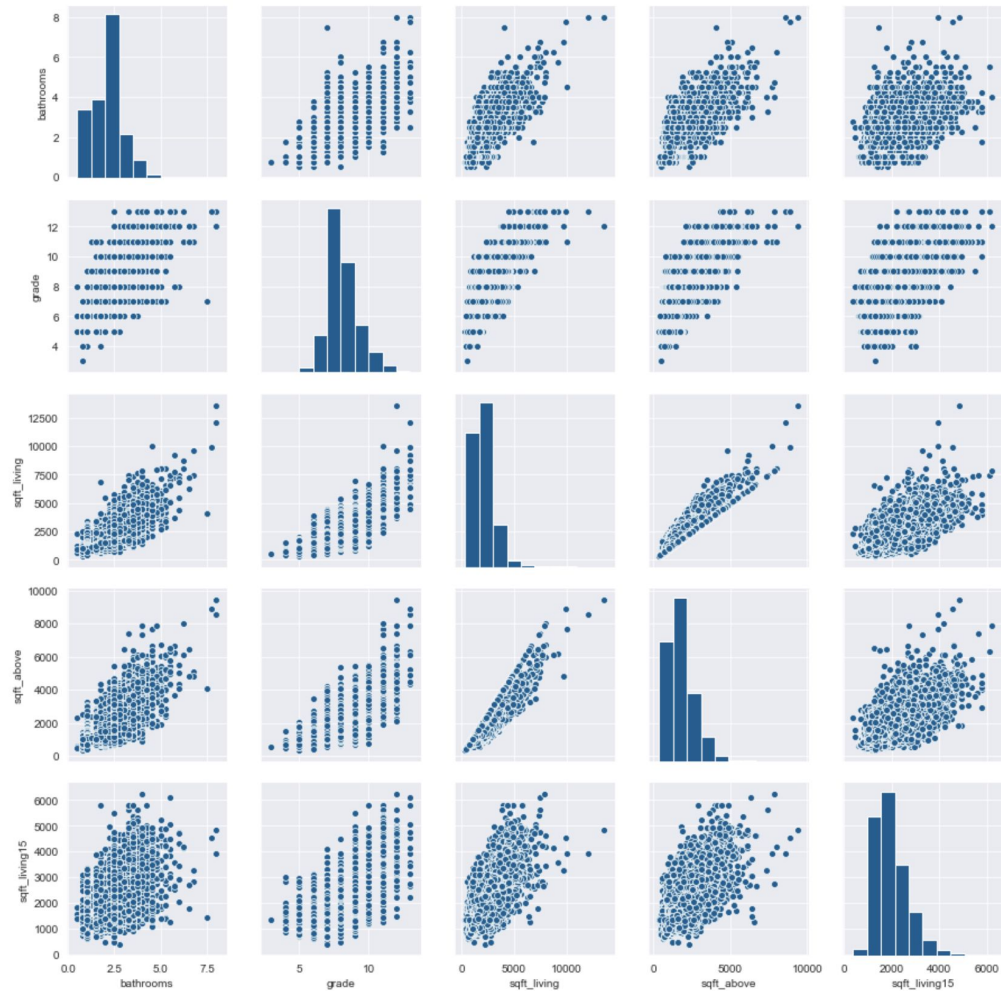
bedrooms_bin = disc_5k.fit_transform(df[['bedrooms']])
bathrooms_bin = disc_3.fit_transform(df[['bathrooms']])
```

Scaling factors:

```
from sklearn import preprocessing
from sklearn.preprocessing import RobustScaler, StandardScaler

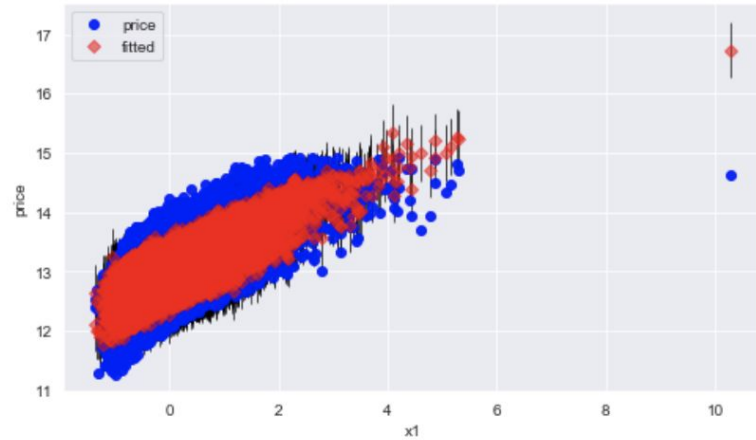
scale_robust = RobustScaler(copy=False, quantile_range=(25.0, 75.0), with_centering=True,
                             with_scaling=True)

scale_std = StandardScaler(copy=False)
```

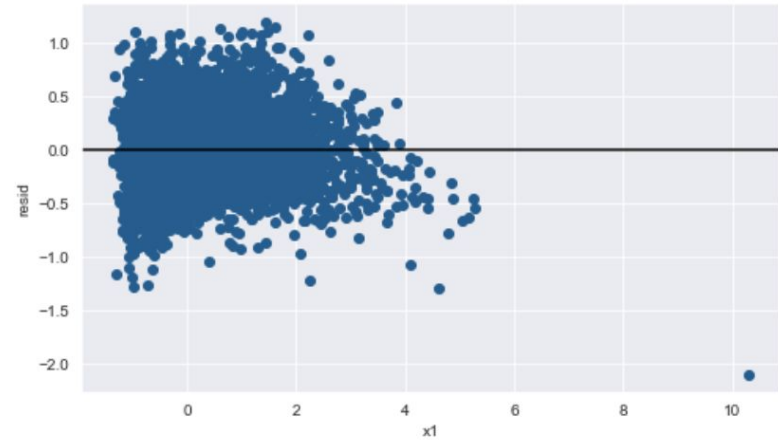


Regression Plots for x1

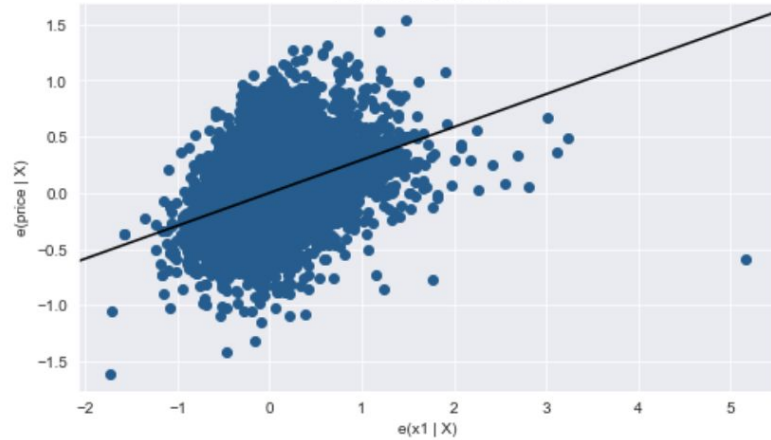
Y and Fitted vs. X



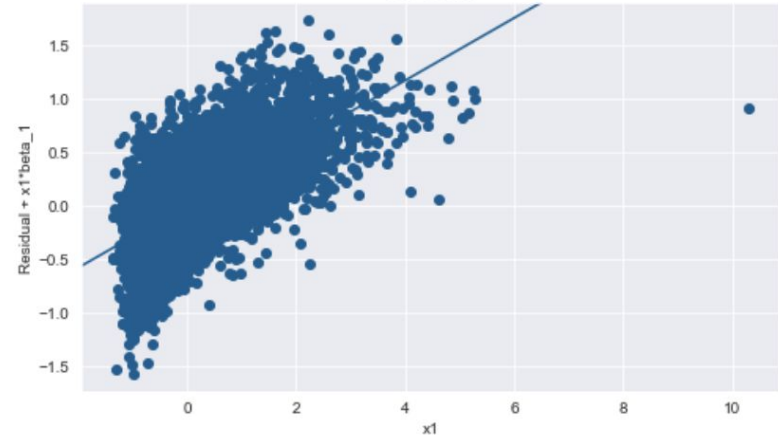
Residuals versus x1




Partial regression plot



CCPR Plot







| | | | |
|--------------------------|------------------|----------------------------|--------|
| Dep. Variable: | price | R-squared: | 0.787 |
| Model: | OLS | Adj. R-squared: | 0.787 |
| Method: | Least Squares | F-statistic: | 2231. |
| Date: | Wed, 08 May 2019 | Prob (F-statistic): | 0.00 |
| Time: | 10:50:44 | Log-Likelihood: | 130.69 |
| No. Observations: | 12068 | AIC: | -219.4 |
| Df Residuals: | 12047 | BIC: | -64.02 |
| Df Model: | 20 | | |
| Covariance Type: | nonrobust | | |

```
mymean = np.full((len(y_test), ), np.mean(y_train))
print("MSE: ", metrics.mean_squared_error(y_test, mymean))
print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_test, mymean)))
rmse = np.sqrt(metrics.mean_squared_error(y_test, mymean))
```

MSE: 0.2660806452187228

RMSE: 0.5158300545903881

| | | |
|---|--|-------------|
|  <code>setup.py</code> | Require <code>scikit-learn>=0.15.0</code> . Resolves #49 . | 3 years ago |
|  <code>tox.ini</code> | added <code>pandas 0.22</code> and <code>sklearn 0.19</code> to <code>tox</code> . | a year ago |

README.rst

Sklearn-pandas

 PASSED

This module provides a bridge between [Scikit-Learn](#)'s machine learning methods and [pandas](#)-style Data Frames.

In particular, it provides:

1. A way to map `DataFrame` columns to transformations, which are later recombined into features.
2. A compatibility shim for old `scikit-learn` versions to cross-validate a pipeline that takes a `pandas DataFrame` as input. This is only needed for `scikit-learn<0.16.0` (see [#11](#) for details). It is deprecated and will likely be dropped in `sklearn-pandas==2.0`.
3. A couple of special transformers that work well with pandas inputs: `CategoricalImputer` and `FunctionTransformer`.

Installation

You can install `sklearn-pandas` with `pip`:

```
# pip install sklearn-pandas
```

Tests

The examples in this file double as basic sanity tests. To run them, use `doctest`, which is included with python:

```
# python -m doctest README.rst
```




Takeaways

- In this dataset, `sqft_living` and `grade` are the variables most highly correlated with price.
- R-squared and MSE/RMSE can diverge sharply, and multiple metrics of accuracy must be taken into account.
- The choice of which variables to keep and which to eliminate for multicollinearity has significant impacts on the accuracy of the final model.
- Data exploration and cleaning take up more time than running the actual regressions, but are crucial precursors to manipulating and processing the information.
- Pandas and scikitlearn contain many powerful tools with extensive documentation to aid in statistical analysis of data, but these libraries present a surprising number of incompatibilities that must be resolved via ad-hoc methods (or discovery of additional libraries like `sklearn-pandas`).