Machine-Learning Analysis:

I used linear regression techniques and random forest regressions to analyze residential housing prices.

For linear models, log base 10 transformations with ridge regression improved the baseline model. Subnbhd was the only geographical variable in the baseline models. The best model with outliers had high dimensional geographical variables removed and one collinear variable removed. Removing high VIF variables did not improve the performance. The model without outliers performed the best with the same variables removed.

BEST LINEAR MODELS FOR GIVEN SCENARIOS:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type | Variables removed | MAPE | RMSE | R^2 |
| Baseline | 'QUADRANT', 'WARD', 'NBHD', 'CENSUS\_BLOCK', 'ZIPCODE', 'SQUARE' | 43.17 | 227499.7660 | 0.6999 |
| Improved baseline  -ridge regression  -log(‘PRICE’) | 'QUADRANT', 'WARD', 'NBHD', 'CENSUS\_BLOCK', 'ZIPCODE', 'SQUARE' | 2.08 | 0.1641 | 0.7587 |
| Best model with outliers  -ridge regression  -log(‘PRICE’) | 'KITCHENS', 'SUBNBHD', 'CENSUS\_BLOCK', ‘SQUARE’ | 2.01 | 0.1587 | 0.7745 |
| Best model without outliers  -log10(‘PRICE’)  -Ridge Regression | 'KITCHENS', 'SUBNBHD', 'CENSUS\_BLOCK', 'SQUARE' | 1.79 | 0.1291 | 0.8406 |

* MAPE- Mean Absolute Percentage Error
* RMSE- Root Mean Squared Error

**Outlier Descriptions (1.7% of data):**

* higher prices
* sold during different times
* lower landarea
* more built in 1940s-1960s
* remodeled 2005-2010
* fewer rooms/bedrooms
* more in NE, Wards 2/6
* Columbia Heights, Petworth, Brookland, Dearwood, Chevy Chase, Mount Pleasant, Congress Heights

**Linear model assumptions:**

Best model with outliers:

-Linear

-Not normal distribution (fat tails)

-Heteroskedastic (small issue)

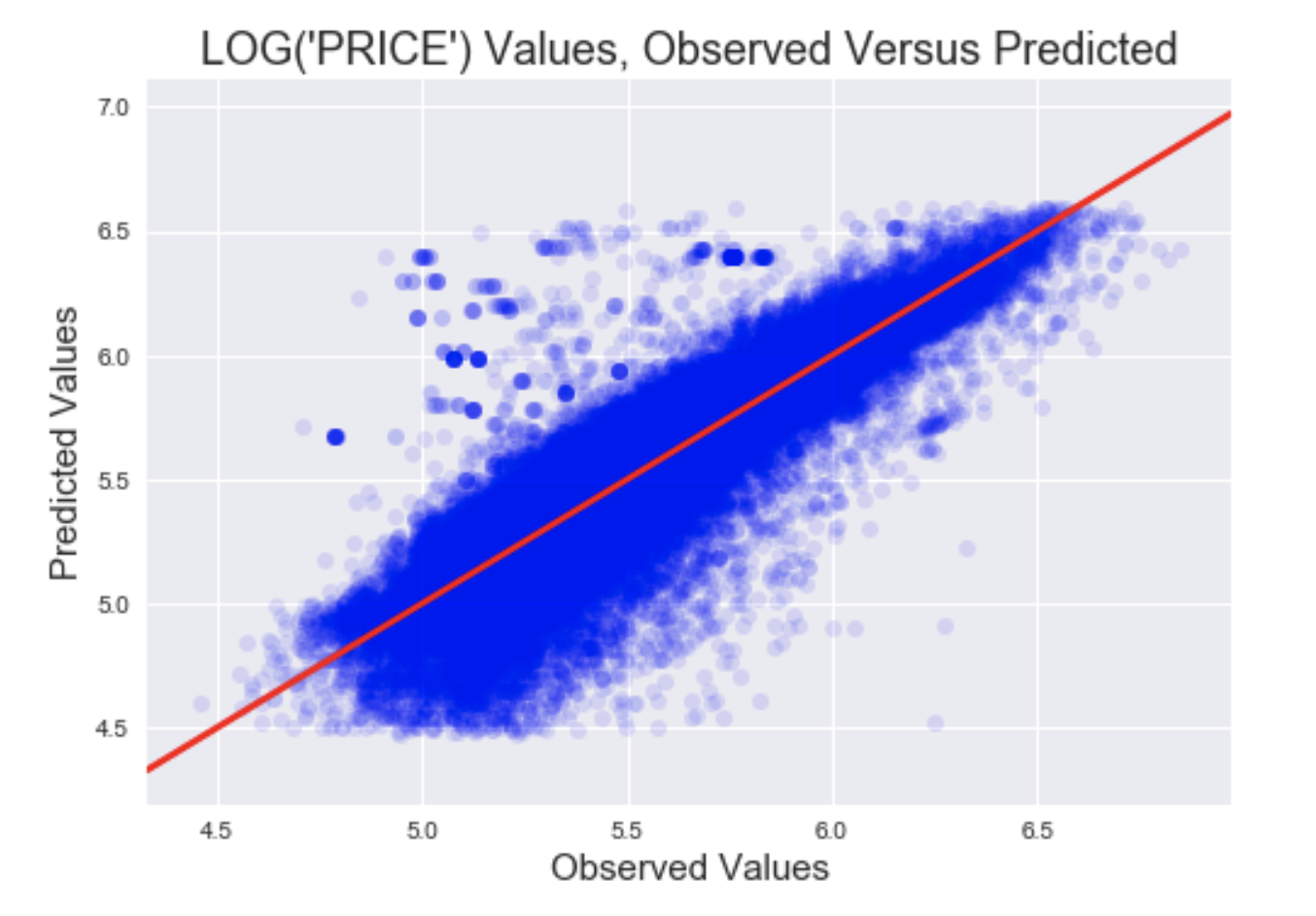
Best model without outliers:

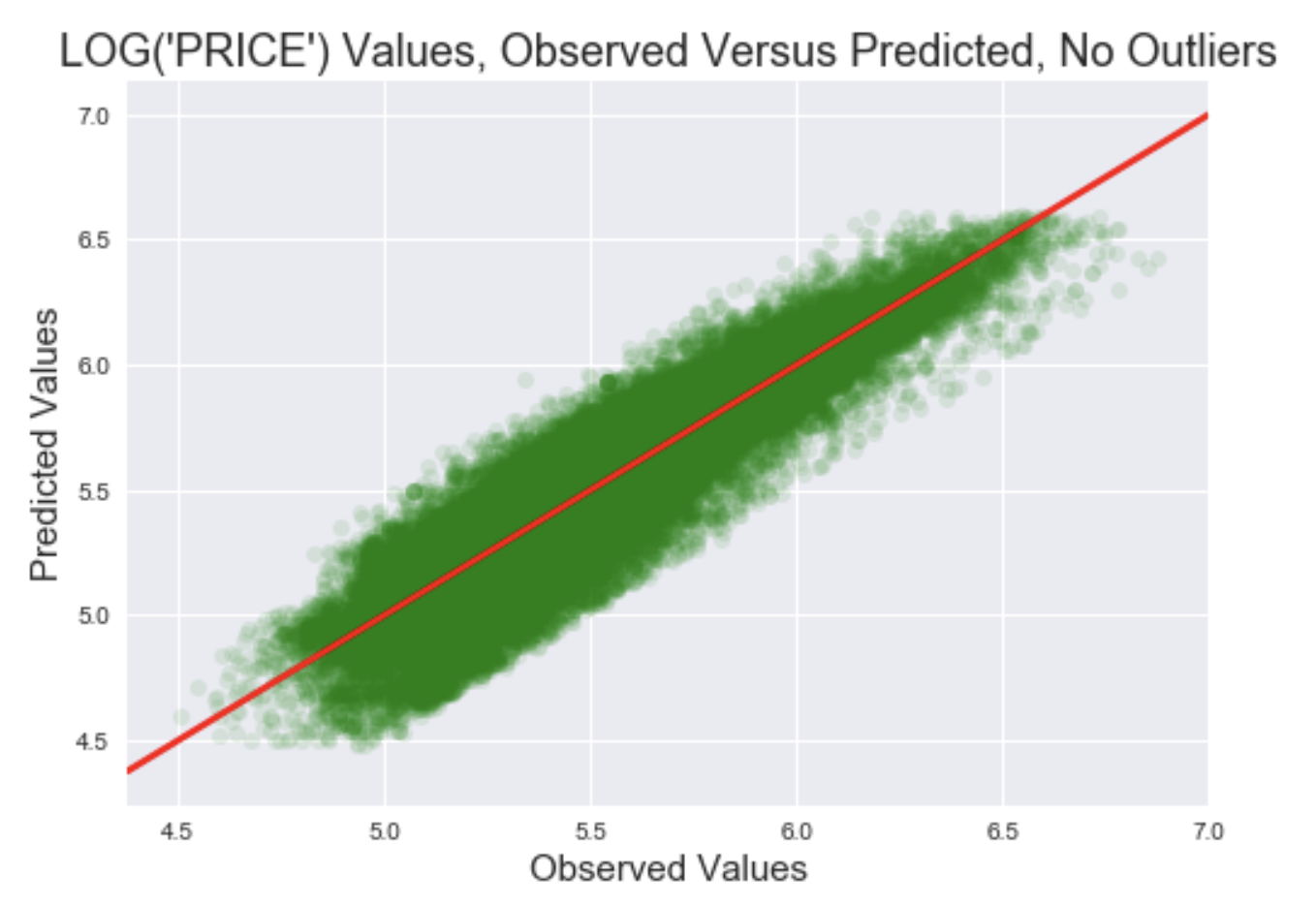
-Linear (small issue)

-Closer to normal distribution

-Heteroskedastic (small issue)

Because our data doesn’t fit some assumptions for linear models, perhaps nonlinear models will perform better.





BEST RANDOM FOREST MODELS:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type | FEATURES REMOVED | | MAPE | RMSE | Parameters |
| Baseline | 'CENSUS\_BLOCK', 'SQUARE' | | 19.3547 | 144251.99 | n\_estimators = 10 |
| Baseline, log | 'CENSUS\_BLOCK', 'SQUARE' | | 1.2550 | 0.11 | n\_estimators = 10 |
|  | | -‘EXTWALL’ | 1.2553 |  |  |
|  | | -‘INTWALL | 1.2554 |  |  |
|  | | -‘SALE\_NUM’ | 1.2550 |  |  |
| **Optimized Baseline log** | 'CENSUS\_BLOCK', 'SQUARE' | | 1.22 | 0.11 | |  | | --- | | bootstrap=True,  criterion='mse',  max\_depth=80, | | max\_features='auto',  max\_leaf\_nodes=None, | | min\_impurity\_decrease=0.0,  min\_impurity\_split=None, | | min\_samples\_leaf=2,  min\_samples\_split=2, | | min\_weight\_fraction\_leaf=0.0,  n\_estimators=110,  n\_jobs=None, | | oob\_score=False,  random\_state=None,  verbose=0,  warm\_start=False) | |
| AdaBoost log | ''CENSUS\_BLOCK', 'SQUARE' | | 3.02 | 0.22 | base\_estimator=None,  learning\_rate=0.05,  loss='exponential',  n\_estimators=90,  random\_state=None |
| GradientBoost  log | ''CENSUS\_BLOCK', 'SQUARE' | | 1.51 | 0.13 | alpha=0.9,  criterion='friedman\_mse',  init=None,  learning\_rate=1,  loss='ls',  max\_depth=3,  max\_features=None,  max\_leaf\_nodes=None,  min\_impurity\_decrease=0.0,  min\_impurity\_split=None,  min\_samples\_leaf=1,  min\_samples\_split=2,  min\_weight\_fraction\_leaf=0.0,  n\_estimators=150,  n\_iter\_no\_change=None,  presort='auto',  random\_state=None,  subsample=1.0,  tol=0.0001,  validation\_fraction=0.1,  verbose=0,  warm\_start=False |

Least influential variables:

EXTWALL, INTWALL, EXTWALL, SALE\_NUM, STYLE, ROOF, HEAT, CNDTN

Removing the least influential variables decreased model performance slightly.

The Random Forest Optimized Baseline model performed the best at predicting the log(Price) with a 1.22 average percentage error rate and 0.11 RMSE. The model also outperformed all linear models.

* We can predict price with a mean absolute percentage error of 17.27%.
* 80% of the time we can predict price between 80% and 125% of its value.

