Web Scraping/Regression Presentation Slides

Predicting Final Sales Price for Single Family Homes in the West San Fernando Valley

George Pappy - 15 September 2021

METIS®

Introduction

METIS®

- Motivation:
  - Historic rise in home sales during the COVID-19 pandemic
  - Prices are at all time highs and still climbing
- Local realtors want a reasonably accurate price predictor
  - Provides a competitive advantage
  - Prices changing too fast: lack of confidence in their own estimates

Methodology

- Primary Data: Single Family Home Sales (Past 90 Days)
  - Excluded Condominiums and Multi-Family Dwellings (different market)
  - West San Fernando Valley only (18 zip codes covering 13 communities)
  - Data scraped from a well-known real estate website (10 predictors):

    Target: Sale Price

    Predictors: Beds, Baths, Square Footage, Lot Size, Year Built, Zipcode, Pool, Garage, Number of Stories (Floors), Average Schools Rating

Methodology (con't.)

- <u>Supplemental Data</u>: Downloadable csv (same website):
    - Homeowners Association Fee (HOA, monthly)
    - Number of Days on Market (could be an alternate target, NOT a predictor)

Methodology (con't.)

METIS®

- <u>Models</u>: Linear Regression with & without Regularization
  - Lasso, Ridge, ElasticNet
  - Also tried tree-based regressors (Random Forest, XGBoost)
  - Extensive feature engineering to improve performance
- <u>Metrics</u>
  - Primarily Mean Absolute Error (MAE, $)
  - Also, R^2 and Root Mean Square Error (RMSE, $)

Methodology (con't.)

- <u>Tools</u>
  - **Requests/BeautifulSoup**: web scraping
  - **Pandas**: clean, explore, engineer features and generate final modeling data
  - **Statsmodels/ScikitLearn**: build regression models as well as to perform cross validation, variable selection and regularization
  - **Matplotlib/Seaborn**: visualizing data exploration, modeling and final results
  - **Python 3.8**: to run all of the above

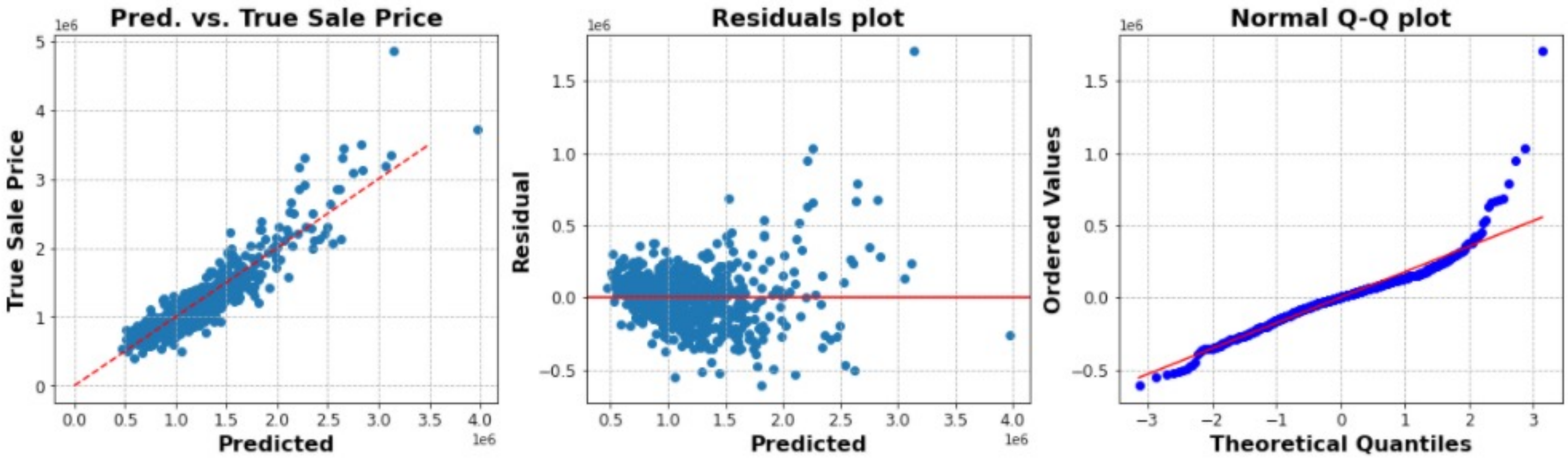
Cross Validation Results – (Model Selection)

- <u>Initial Linear Model (Baseline)</u>

  (11 predictors → 26 predictors after one-hot encoding the distinct zipcodes)

```
Basic Linear Model: Mean CV R-squared = 0.826 +/- 0.015
Basic Linear Model: Mean CV MAE = $136193 +/- $10081
Basic Linear Model: Mean CV RMSE = $197568 +/- $27254
```

Should be able to do better than this



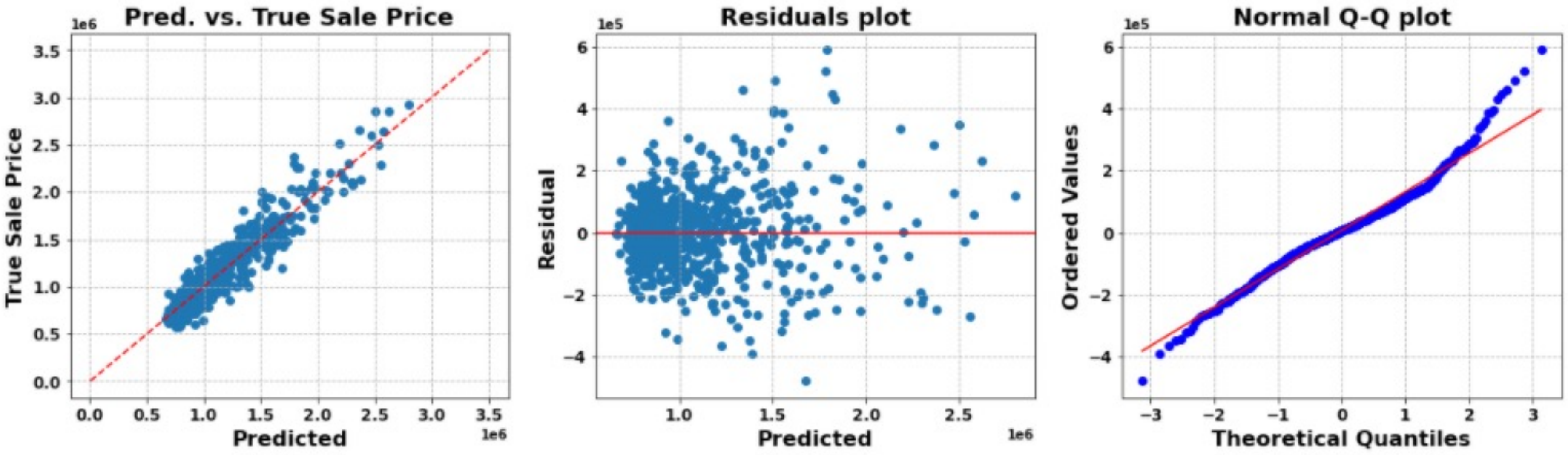
Cross Validation Results – (Model Selection)

- <u>Best Linear Model: ElasticNet</u>

  All 2nd-order terms & interactions filtered down to 58 Lasso-selected predictors; log(target)

```
Basic Linear Model: Mean CV R-squared = 0.845 +/- 0.023
Basic Linear Model: Mean CV MAE = $101397 +/- $6325
Basic Linear Model: Mean CV RMSE = $141169 +/- $12635
```

Much better!

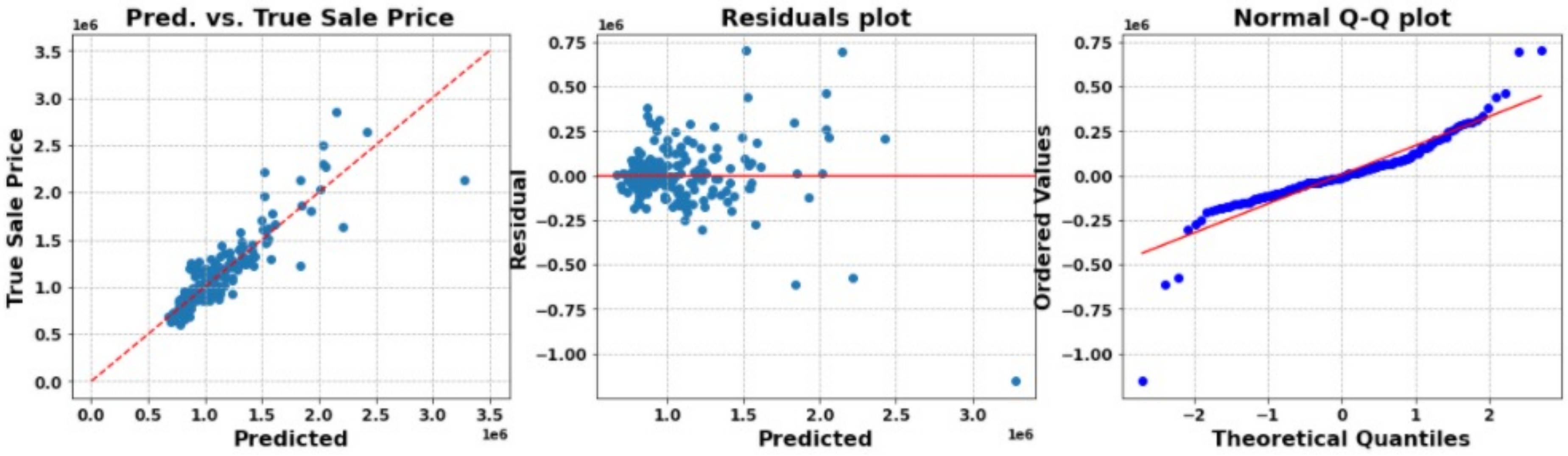


Test Set Results – Final Selected Model

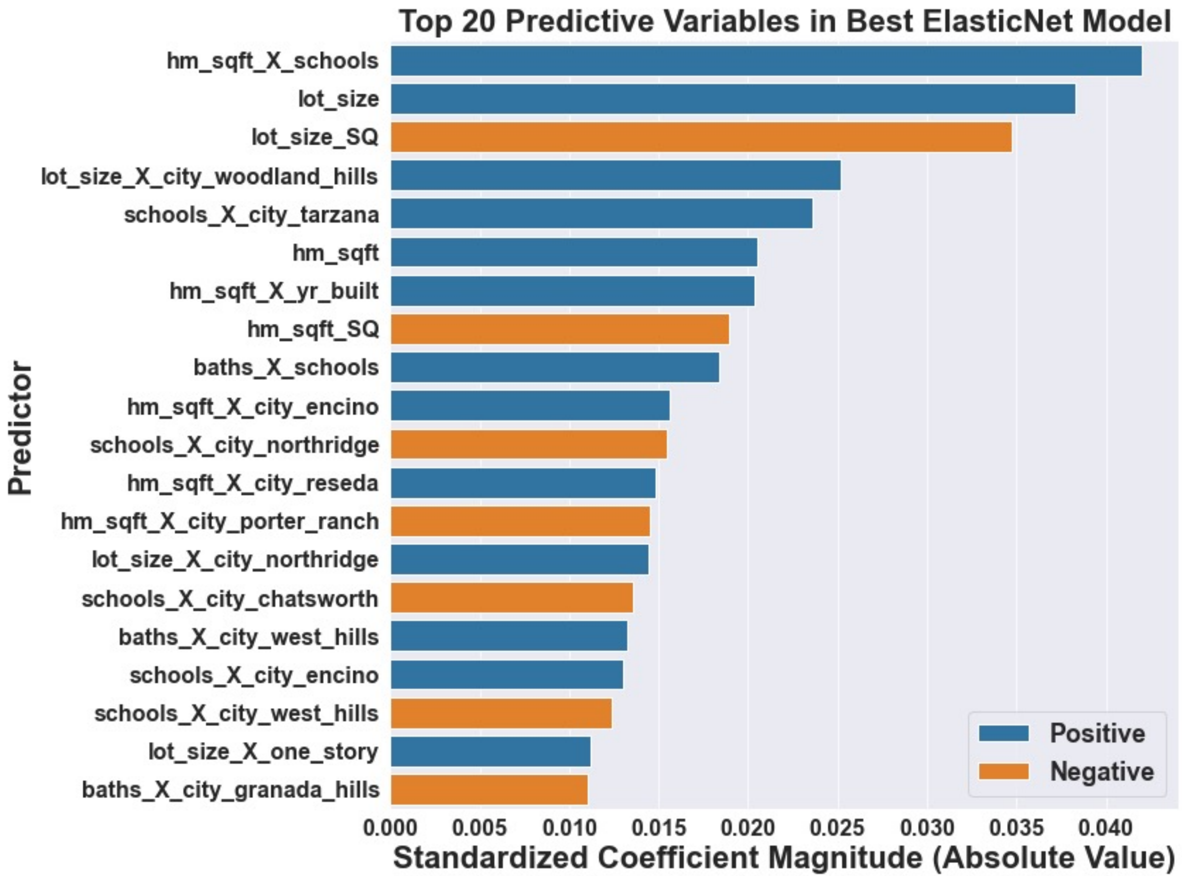
- **Best Linear Model: ElasticNet**

All 2nd-order terms & interactions filtered down to 58 Lasso-selected predictors; log(target)

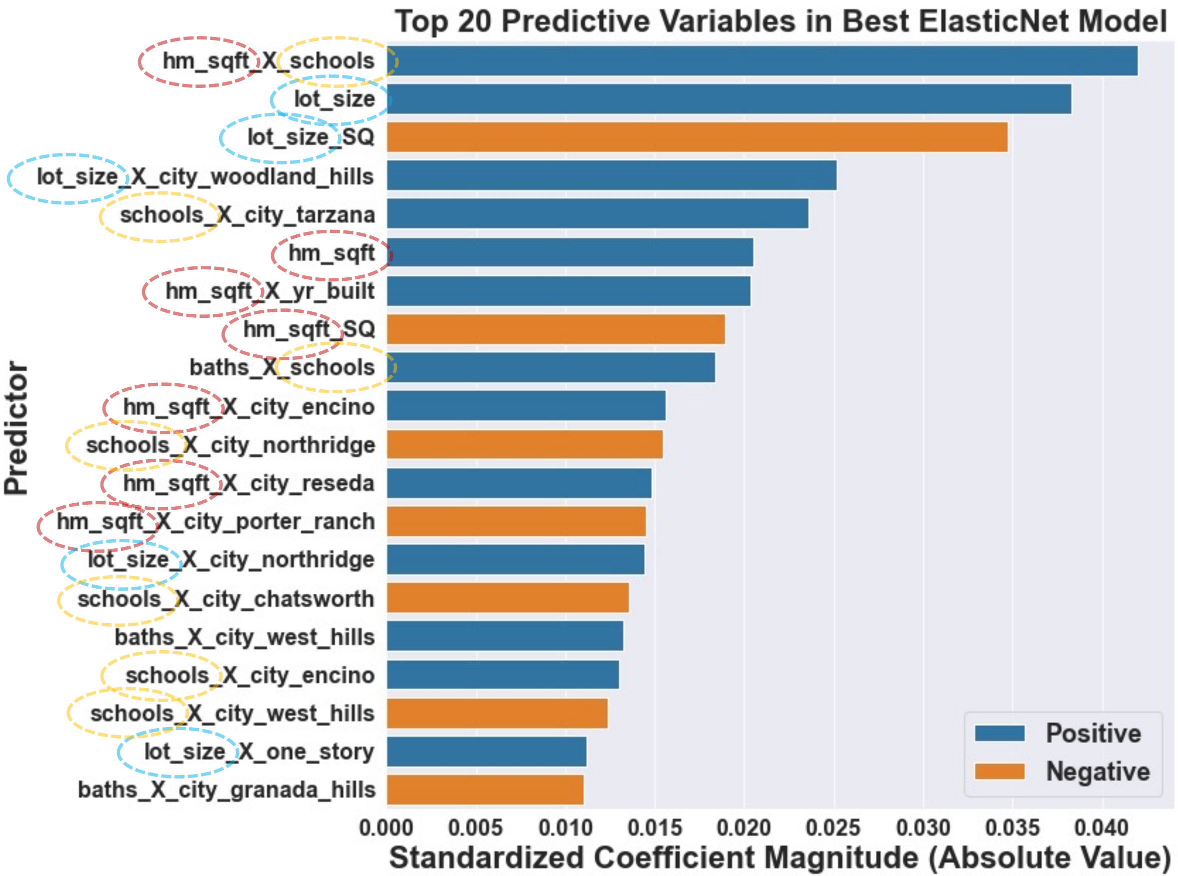
```
Best Linear Model (ElasticNet): Test Set R-Squared = 0.82
Best Linear Model (ElasticNet): Test Set MAE = $112550
Best Linear Model (ElasticNet): Test Set RMSE = $177272
```



Test Set Results – Final Selected Model



Test Set Results – Final Selected Model



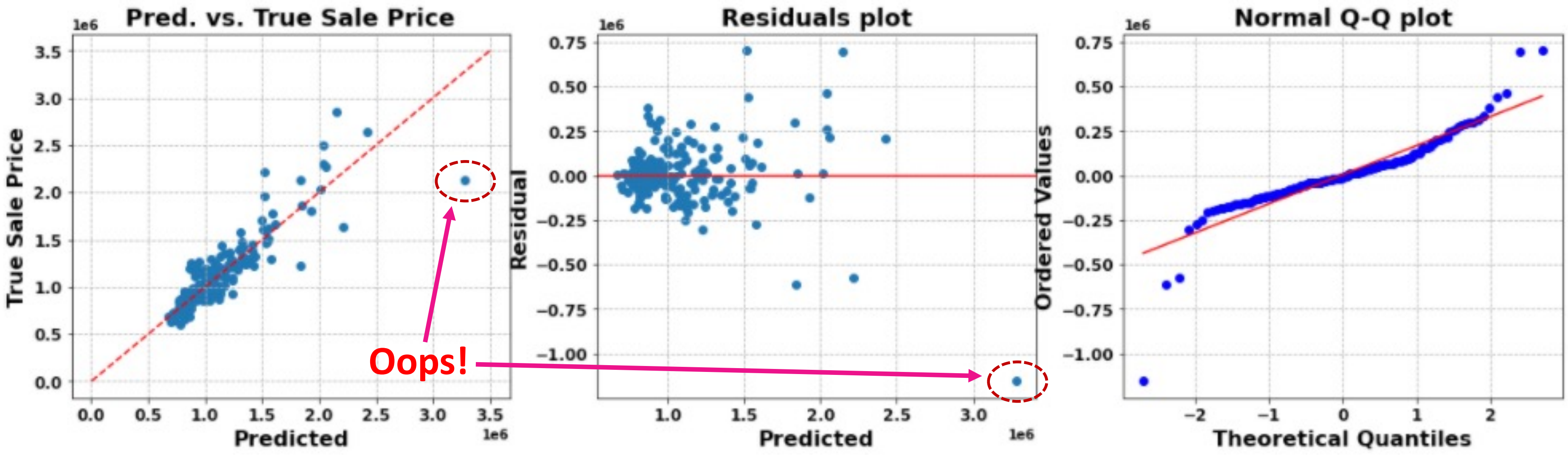
- Square Footage, Lot Size & Schools Rating play very important roles in this model's predictions
- Initial Exploratory Data Analysis (EDA) showed high correlation to the target for Square Footage and Lot Size
- Schools Rating had much weaker target correlation (0.36), so this is an interesting result

Test Set Results – Final Selected Model

- Best Linear Model: ElasticNet

All 2^{nd}-order terms & interactions filtered down to 58 Lasso-selected predictors; log(target)

```
Best Linear Model (ElasticNet): Test Set R-Squared = 0.82
Best Linear Model (ElasticNet): Test Set MAE = $112550
Best Linear Model (ElasticNet): Test Set RMSE = $177272
```

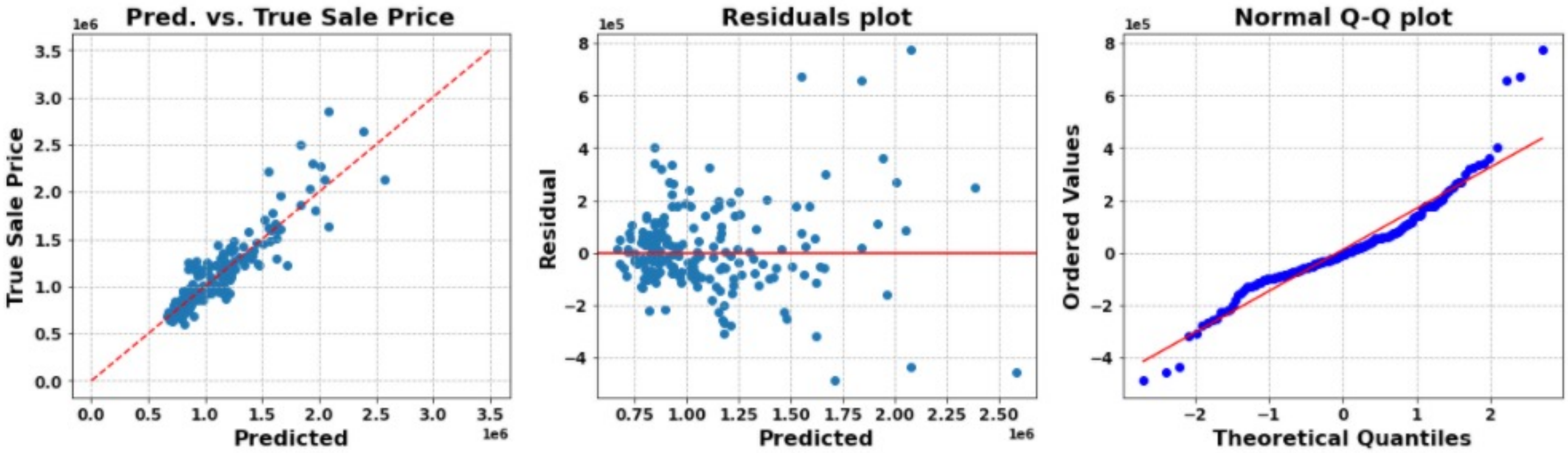


Test Set Results – Other Final Selected Model

- <u>Best Tree-Based Model: XGBoost</u>

All 2nd-order terms & interactions filtered down to 58 Lasso-selected predictors; log(target)

```
XGBoost: Mean CV R-squared = 0.832 +/- 0.032; Test Set R-Squared = 0.82
XGBoost: Mean CV MAE = $105136 +/- $9840;  Test Set MAE = $111591
XGBoost: Mean CV RMSE = $148594 +/- $16258; Test Set RMSE = $163912
```



Conclusions

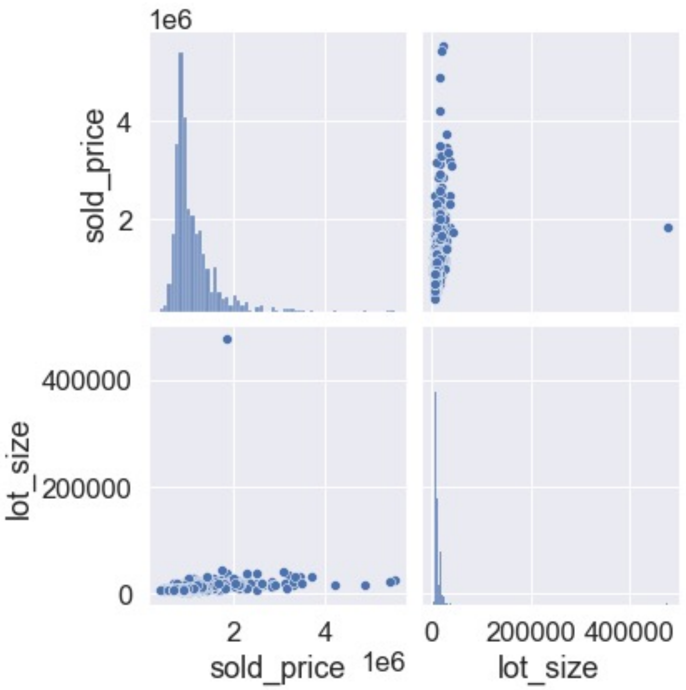
METIS®

- Recommendations
  - If possible, use the XGBoost model in the short term
  - Longer-term, seriously consider funding the future work (see below)
- Interesting Insights
  - Square Footage, Lot size, School Rating are the most important predictors
  - Square Footage is a proxy for Beds and Baths
- Future Work
  - Create a **Home Rating** categorical predictor (e.g., {Poor, Fair, Good, Excellent})
  - Train, cross validate & test with **more data** (maybe scrape the entire Valley)
  - Consider **narrowing the price range down** even more (perhaps <= $2Mil)
  - Build a predictive model for the **alternate target**: Number of Days on Market

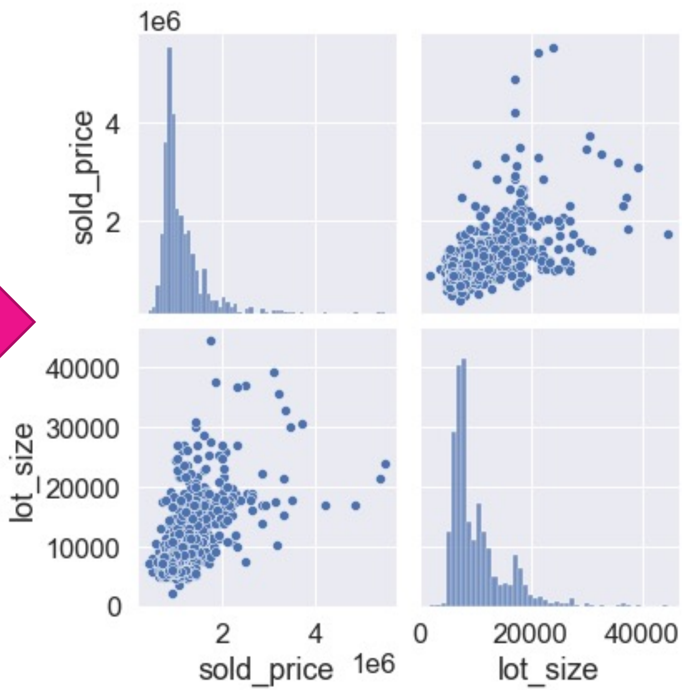
Appendix

One Obvious Initial Outlier Was Dropped

- One home (out of 1011) had a Lot Size more than 10x that of any other



Drop the outlier



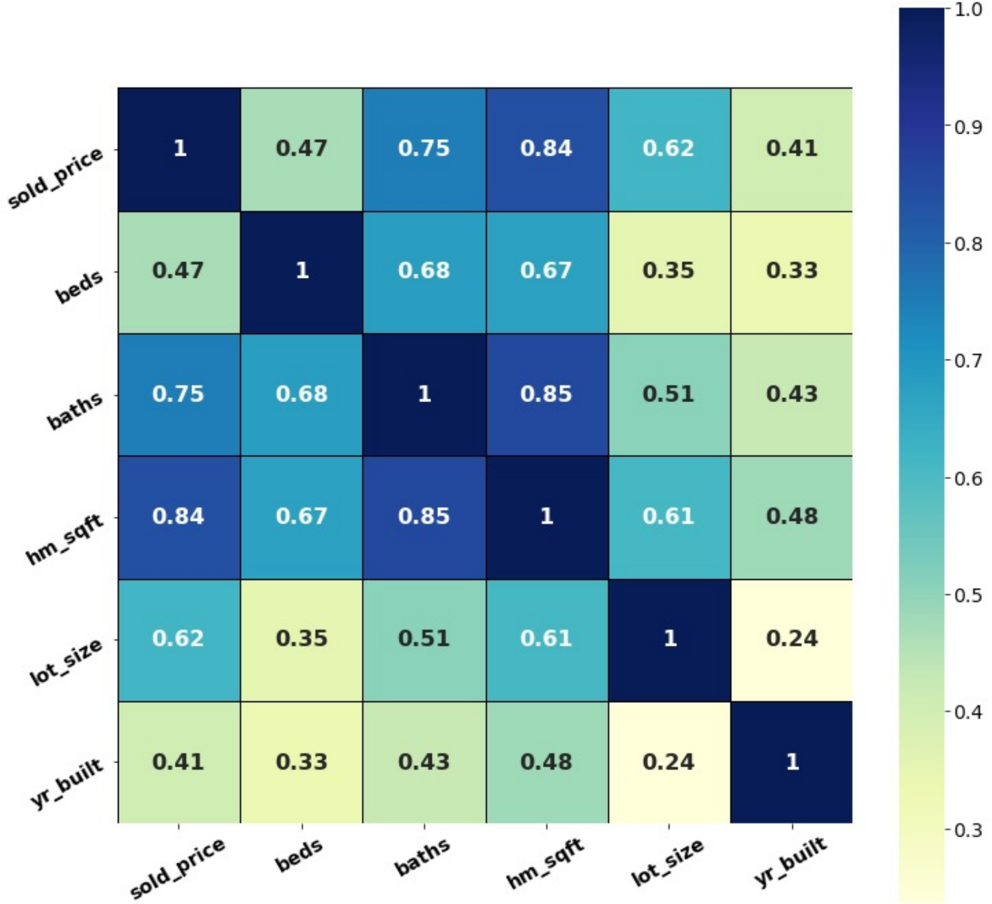
Distribution of Sale Prices in Dataset

- Dropping prices outside of [$575k, $3.0M] is justified by this distribution (even dropping prices ≥ $2.0M is probably reasonable)

	sold_price
count	1.010000e+03
mean	1.125490e+06
std	5.020166e+05
min	4.000000e+05
25%	8.400000e+05
50%	9.650000e+05
75%	1.260000e+06
max	5.500000e+06

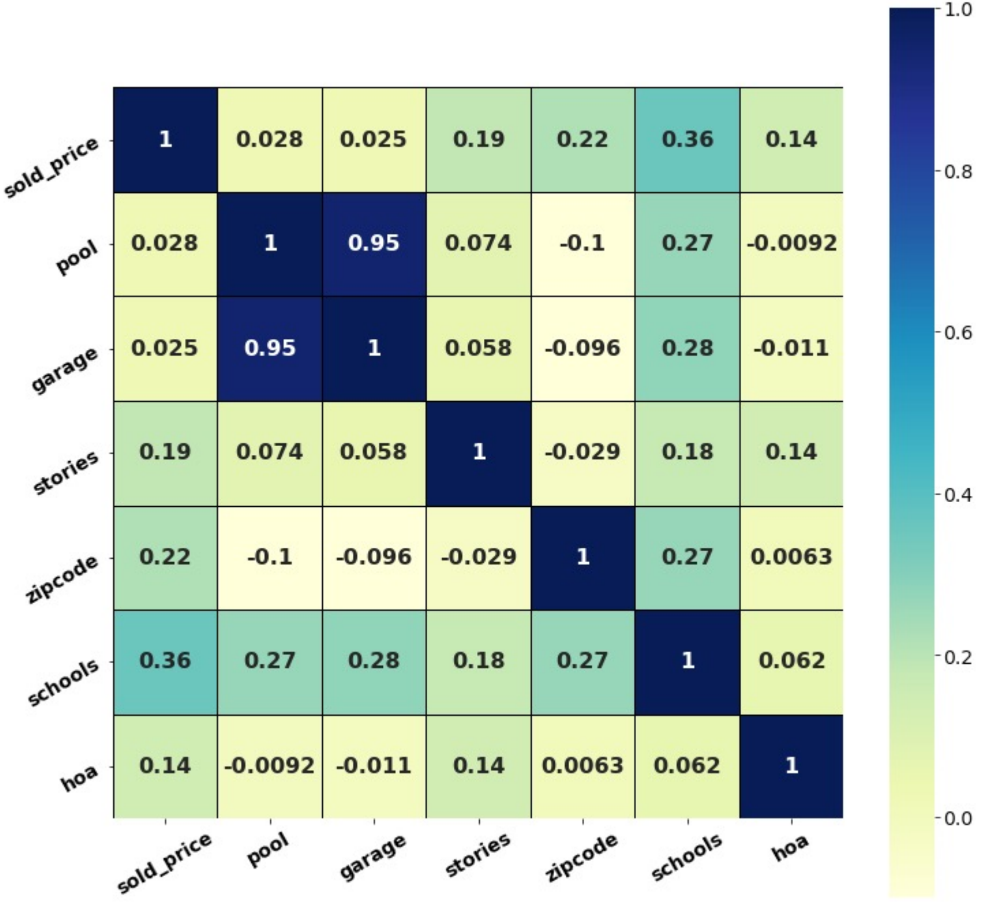
Original Data Correlations w/ Sales Price

- Square Footage, Baths and Lot Size are by far the strongest correlations
- Note strong correlations between:
  - Beds & Baths
  - Sq. Footage & Beds
  - Sq. Footage & Baths



Original Data Correlations w/ Sales Price

- Schools Rating has a fairly weak correlation with the target
- All other correlations to the target are weak or very weak
- <u>Note</u>: Pool and Garage are highly correlated with each other, leading to the decision to drop Pool from the modeling dataset



METIS®

No Real VIF Issues w/ Beds/Baths/Sq. Feet

METIS®

- Despite strong correlations between Beds, Baths & Square Feet, their VIFs are not really an issue
- Decision was made to accept VIF of 5.44 for Square Feet (barely > 5.0)
- No surprise that Pool & Garage have such high VIFs given 0.95 correlation (justifying decision to drop Pool from the dataset)

Variable	VIF
beds	2.374858
baths	4.072735
hm_sqft	5.437265
lot_size	1.873326
yr_built	1.570090
pool	11.575024
garage	11.747186

High VIFs in One-Hot Encoded Zipcodes

METIS®

- Variance Inflation Factors (VIFs) surprisingly high for the Zipcodes
- Ideally want all VIFs < 5 (as is true for the other variables)
- Idea: try reducing 18 Zipcodes down to 13 communities (some of which contain multiple Zipcodes)

Variable	VIF
zipcode_91303	6.572945
zipcode_91304	17.114664
zipcode_91306	9.160317
zipcode_91307	14.155095
zipcode_91311	9.843143
zipcode_91316	12.274967
zipcode_91324	6.336829
zipcode_91325	10.785262
zipcode_91326	13.688698

Variable	VIF
zipcode_91335	14.904258
zipcode_91343	14.012338
zipcode_91344	23.141578
zipcode_91356	9.990088
zipcode_91364	22.477096
zipcode_91367	19.839412
zipcode_91406	17.563902
zipcode_91436	5.248329

Zipcodes Mapped to Cities: VIFs Much Better!

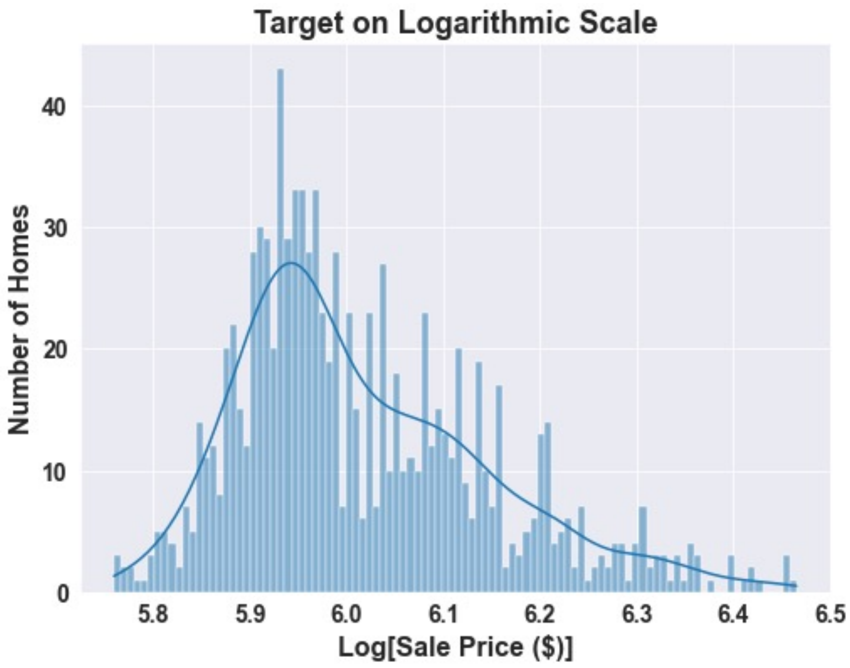
METIS®

- Simply mapping 18 Zipcodes to their 13 associated city (community) names and one-hot encoding those variables solves the high VIF problem!

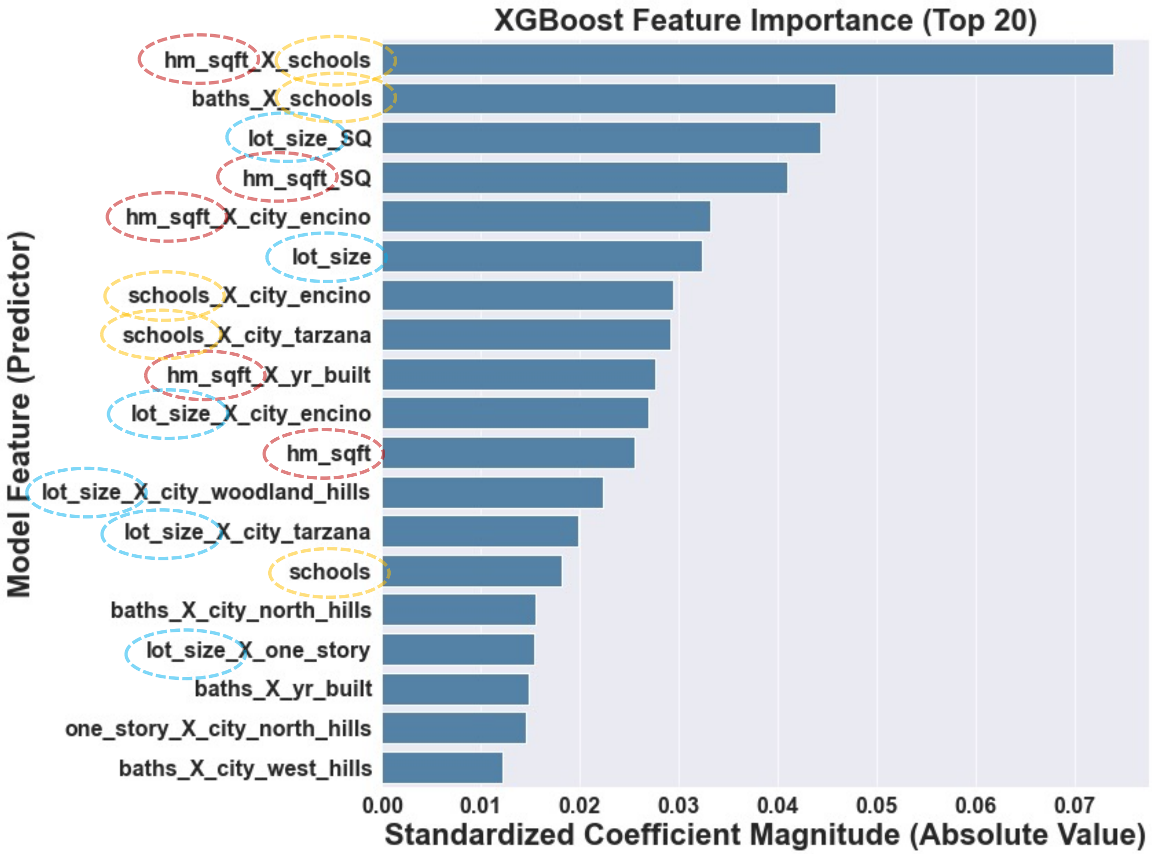
Variable	VIF	Variable	VIF
city_chatsworth	1.461258	city_porter_ranch	2.009986
city_encino	2.162079	city_reseda	1.692616
city_granada_hills	2.372798	city_tarzana	1.645058
city_lake_balboa	1.942892	city_west_hills	1.969172
city_north_hills	1.767004	city_winnetka	1.390593
city_northridge	1.854837	city_woodland_hills	4.226194

Log-Transforming Target Improved Model

- Log-transformed target less skewed, improving model performance



XGBoost Model– Variable Importance



- Square Footage, Lot Size & Schools Rating also play very important roles in this model's predictions
- Initial Exploratory Data Analysis (EDA) showed high correlation to the target for Square Footage and Lot Size
- Schools Rating had much weaker target correlation (0.36), so this is an interesting result

Random Forest Models Never Beat XGBoost

- <u>Best Random Forest Model</u>

  All 2nd-order terms & interactions filtered down to 58 Lasso-selected predictors; log(target)

```
RandomForest: Mean CV R-squared = 0.818 +/- 0.035; Test Set R-Squared = 0.806
RandomForest: Mean CV MAE = $109312 +/- $8352;  Test Set MAE = $113709
RandomForest: Mean CV RMSE = $155945 +/- $16837; Test Set RMSE = $174812
```