**San Diego House Price Predicting**

**Wen Yan, Mengting Wang, Salah Ahmad, Xia Song**

### Data Outliers Detection

In order to improve data quality and prediction accuracy, boxplot method were employed to detect the outliers of target prediction variable (square feet price). Fig 1 showed boxplot of square feet price in zip code level, and some outliers were found in each zip areas. These outliers may cause bias when comparing the observation data and prediction data. In order to reduce these bias, we filter out all the outliers and the cleaned data looks much better. The outlier criteria is 1.5 IQR outside of Hinge-spread based on boxplot.

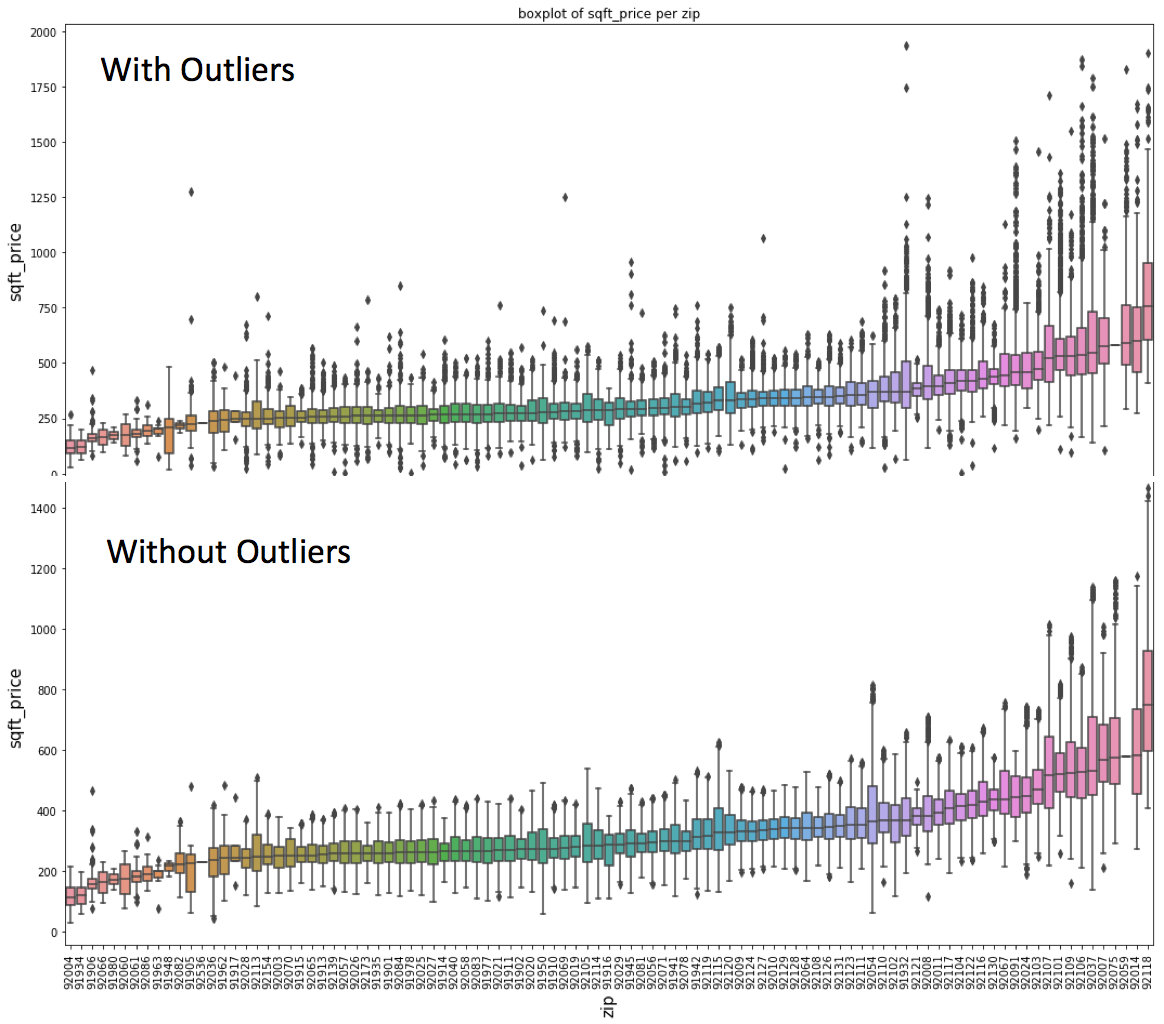
****

Figure 1. Sold square feet price outlier detection

### Model Result Interpreting

Random Forest Model was used to predict house price in San Diego area. The tracking window ('2015-10-01', '2017-10-01') is used to train and tune the baseline model which will be used to calibrate various improvement later on. And from previous section, we can see optimized sliding and testing window are 12 months and 4 months separately.

* Error( RMSE): testing: 127516, training: 93770
* The feature set and importance. Model ranked square feet as the most important feature, and average square feet price in zip code level is another important feature. In addition average sold price in zip code level, improvement over land, longitude and latitude are also obviously contributed to model simulation (Fig 2).

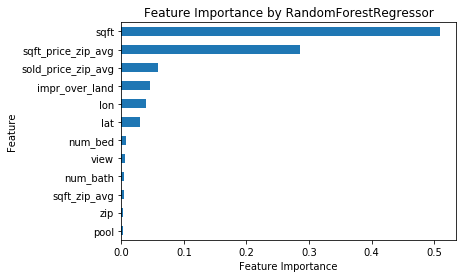


Figure 2. Feature importance

* Learning curve of RandomForestRegressor from the validation ranking out of 75 parameter combinations from grid search. We can see after about 25 iterations, the model improves very slow. Each big bump is when we use bigger max\_depth, and each slow improvement is by using bigger n\_estimators (Fig3).

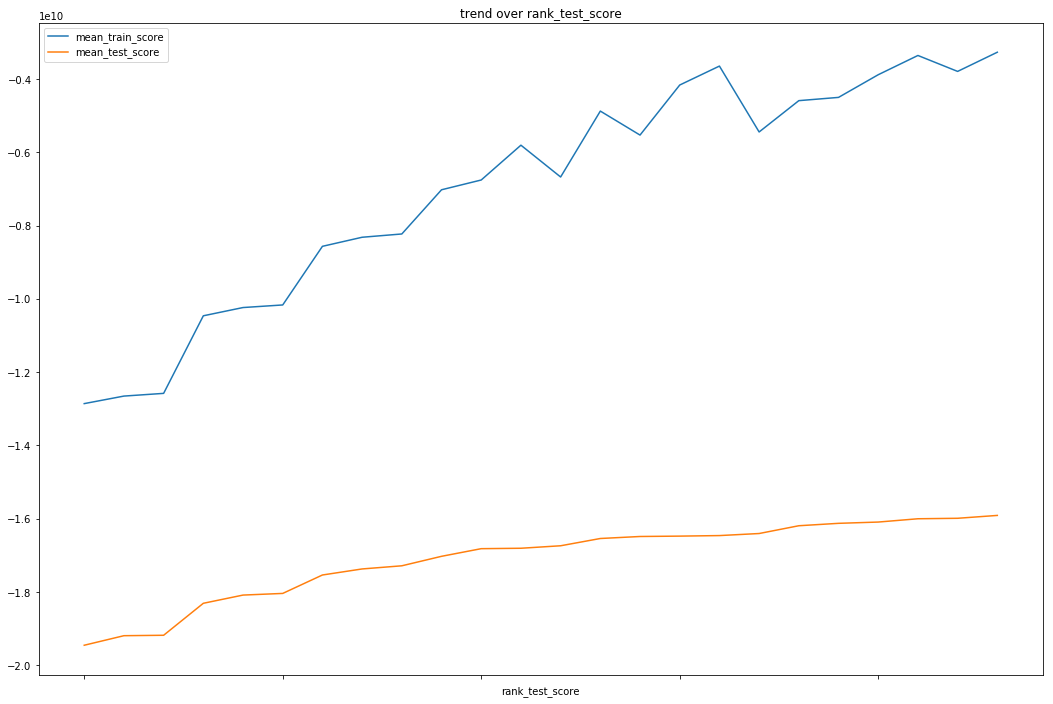


Figure 3. Validation learning curve

From figure 3, we can see RandomForest is pretty resistant to overfitting, so we can use big enough value for max\_depth (eg 14) which will reduce the bias error, while the performance goes flat as we increase n\_estimators and max\_depth, we may want just choose good enough n\_estimators which gives similar performance but saves compute power.

### Model Evaluating

**Evaluation Metrics**: RMSE on target variable ‘sold\_price’

**Residual Analysis**

The baseline model turns to overestimate the cheap houses but underestimate the more expensive houses. Residuals could reach 1M for houses below .5M, which means these big residuals are likely to be outliers, eg foreclosure, gift etc.

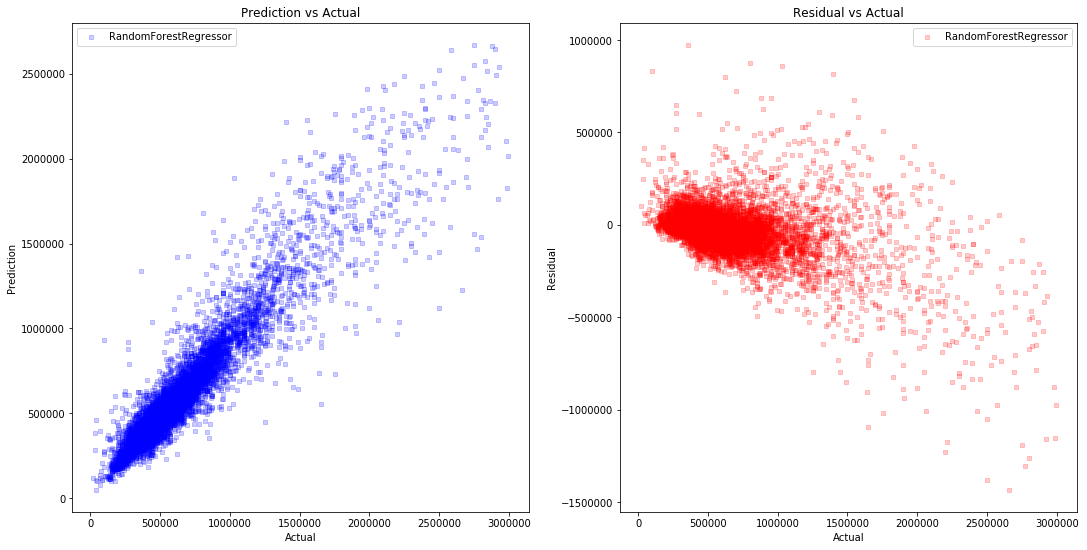


Figure 4. Prediction versus Actual, and Residual versus Actual

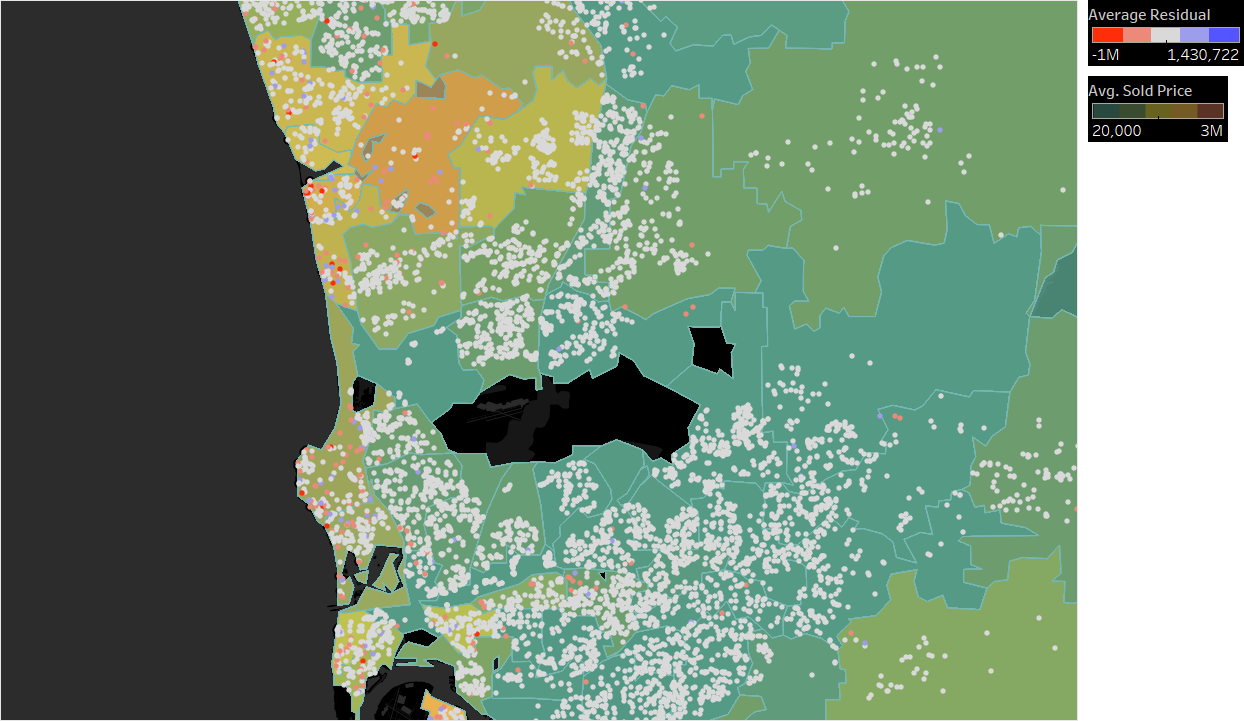


Figure 5. Residual spatial distribution

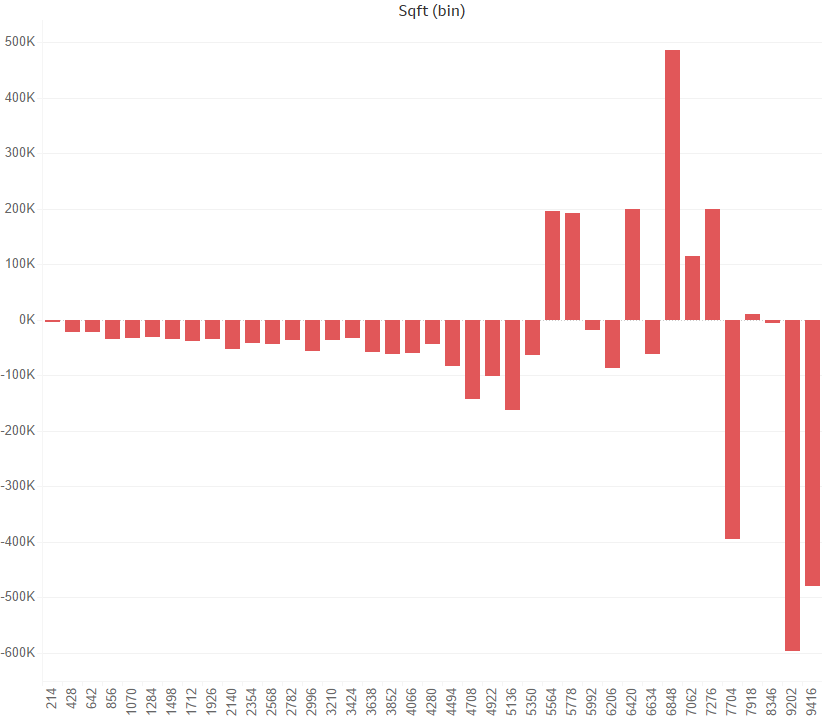
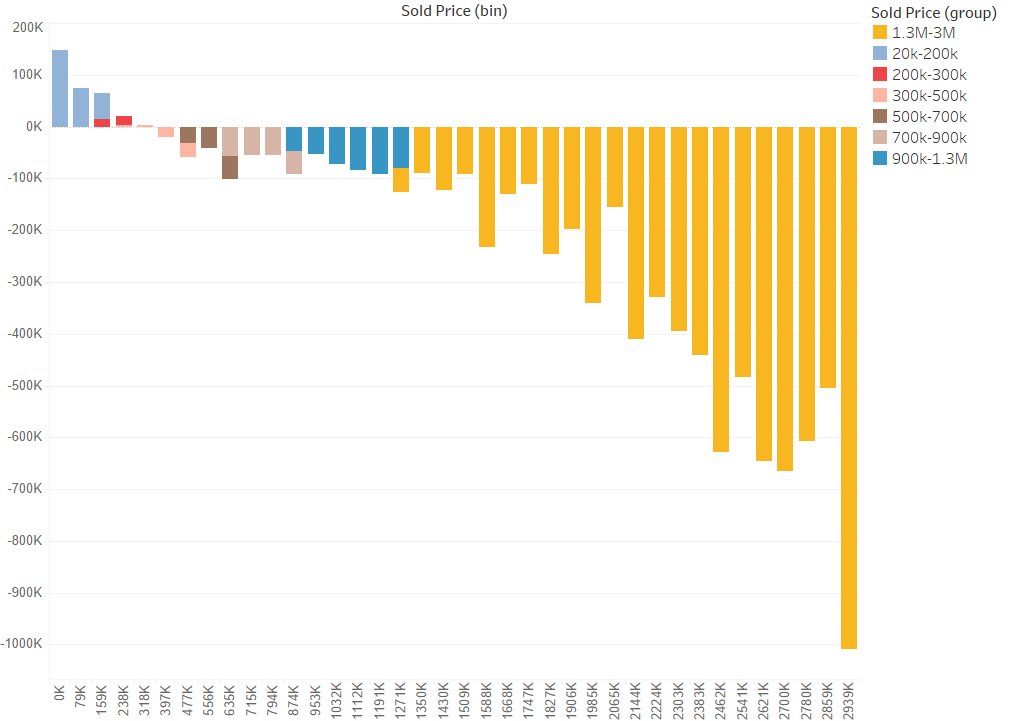


Figure 6. Relationship between residual and sold price, square feet

We developed a set of visualizations to help identify the distribution and trend of residuals. The visualization identifies a clear trend that expensive properties tend to be more underestimated than cheaper ones. From Fig 5 we can see that even though extreme underestimated values are scattered throughout the county, coastal areas tend to have more than others (more clustered). Plotting residuals over sqft segment, a clear trend shows that under ~4300 sqft residuals are relatively flat and low; properties over 4300 sqft tend to have less pattern on residuals and more fluctuations are presented in that range (Fig 6).

### School Feature Engineering

We get 2 new dataset for schools.

1. All San Diego county’s schools location in lon/lat from ‘Sandag’
2. Scraped school ratings of all San Diego county from ‘greatschools’.

The features we engineered so far(still ongoing):

1. **min\_elem\_distance, min\_middle\_distance, min\_high\_distance**: the distance to closest elementary/middle/high school
2. **elem\_rating, middle\_rating, high\_rating**: the rating of closest elementary/middle/high school
3. **avg\_elem\_rating, avg\_high\_rating, avg\_middle\_rating:** the average rating of 3 closest elementary/middle/high schools

The procedure to get above features:

1. First join the 2 datasets by matching school name and address. We are able to match 90% of the schools and ~ 10% of schools can’t be matched.
2. Calculate the distance between each house and each school in same zip and sort the distance from closest to furthest. This is solely completed through Postgis functions.
3. Get top 3 schools of each type for each house to calculate above features.
4. For houses that doesn’t have any school related to, several methods were tried to fill the missing value for each feature. The most promising method is use the max distance for distance features and 0 for ratings which is the least value.

We saw about 2% improvement with these features as below, which is still far below our expectation. The procedure and the code need be reviewed for further improvement.

Baseline model without any other improvement has RMSE: 127690

Baseline with above school features RMSE: 124974

### Multi-Segment Modeling

As suggested in the residual analysis, one single model overestimate lower price segment and underestimate high price segment which suggest different price ranges may behave different due each nature, so multi-segment of modeling is necessary to target on each segment. Below are 2 methodology we tried so far.

**Segment on county’s evaluation (land + improvement)**

Evaluation is known variable for all houses and their transactions, so we run clustering on the evaluation value to get natural segments based on density of this value. This is done on both training and testing dataset. And RandomForestRegressor with same hyperparameters were trained for each segment and used for prediction on according test segment.

Final error and residual for 2 segment modeling is as below.

Table 1. Residuals under different house price scenarios

|  |  |  |  |
| --- | --- | --- | --- |
| Scenarios | All house price | House price < 1 M | House price > 1M |
| Residual | 113,057 | 72,421 | 240,146 |

### 

Figure 7. Relationship between residual and sold price, square feet

And Fig 8 below is the plotting of testing error with increasing cluster/segment number.

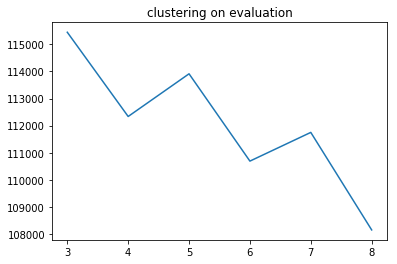


Figure 8: overall RMSE with different segment number

It’s a bit hard to understand the zigzag trend with different segment number, but in overall, the total error keeps dropping. And above 8 segments we face the problem of too few data samples in certain segments, so we can’t keep increase the number. But overall improvement on model performance is above 4%. Using county’s evaluation for segmentation has some limitation because the evaluation can deviate a lot from house’s true value if it doesn’t have any transaction for a long time since it didn’t trigger county’s re-evaluation on this house.

**Segment on sold\_price**

This is tricky since sold\_price is the model’s target, we can’t assume its value for test dataset in advance for clustering, but here’s the methodology we tried although it doesn’t turn out working. We can still do segmentation on sold\_price for training dataset and build multiple models. Then we run all models for test dataset, so each test data get multiple prediction, we use voting method on most possible segment these predicted price fall in and use that segment prediction as final result. Problem we see is a lot of test data can’t get an agreed voting result.

### New Data Obtained and in Pipeline

* Foreclosure data from 1986 till 2018: At Parcel level, foreclosure is tracked by date. Since the Parcel is unique, the data will be linked with the rest of data sets we have such as property characteristics and sales history.
* Accurate Year\_built attribute for vast majority of the properties. Recall the previous issue that we have been suffering from is that we didn’t have good enough year-built information, as all years were given in 2 digits format, therefore it blended in the noise of certain houses being left in ambiguity of their year built (i.e. 00 referring to 1900 or 2000). This use to count for considerable portion of our data. We worked with SD county domain experts and cross-checked historical tax bill datasets and came up with a series of query logics to largely increase the accuracy of this data. Now with accurate year\_built, it can be used as an important feature for our model.
* San Diego Regional Economic Indicator History

We currently obtained monthly data of San Diego Regional Economic Indicator History data from 2012 to 2018, which includes 20+ macro/regional economic factors for San Diego county and nationwide. A couple of important features include:

CPI (Consumer Price Index), DOW, Unemployment Rate, Days on Market (existing single family sales), bankruptcy filings, Average Gas Price, Auto and Motor Vehicle sales, Retail Food Sales, etc.

Considering the magnitude of the project, 6 years worth of data may seem to be insufficient. We’re in the process of communicating with SD county to try to get more of the historical data for this set. If we are able to obtain more than 10 years of data, we will:

* Perform EDA on combined dataset (with current Housing data) to find any correlations
* Include selective features into model and place the model into macro-economic environment to seek for reasonings of certain trend and/or improvement of current model

### Next steps for improvements - Additional Versions on top of the baseline

* Version 2: Segment the data into 2, 3 or 4 (cluster on price only). Use voting method for results evaluation. Capture the results and see if we get better than the baseline model.
* Version 3: Same as Version 2, but cluster based on neighborhood.
* Version 4: Engineer and add the year\_built feature to the model
* Version 5: Bring additional data sources to the model such as: school distance, prime rate/libor, inflation/GDP.

Note: These are only the versions that we have identified and thought about at this time. We expect we will create many others.

### Individual Contribution

Salah: Getting foreclosure data and Year-built

Wen: Feature engineering on school related features

Mengting: Explore macro-economic data for model improvements

Xia: Prepare initial report draft