**Analysis of Zillow Dataset for House Prices**

**Evaluate colleagues’ model predictions**

Assumption: Linear Regression model used for predicting sale prices of the houses

Variables Excluded Round 1: PropertyID, UseCode and Prediction. These are the variables that were not needed for basic correlation matrix and other analysis.

Correlation Matrix: When the correlation matrix was calculated, highly correlated variables were BGMedHomeValue, FinishedSquareFeet. Negative correlated variables with SaleDollarCnt were Longitude, BGPctKids, BGMedYearBuilt indicating that the prices of the house decreased with the decrease in these variables.

After careful observation of the correlation matrix, censusblockgroup had no contribution to assess the sale price of houses. Transdate and ZoneCodeCounty did not contribute either. Therefore,

Variables Excluded Round 2: censusblockgroup, Transdate, ZoneCodeCounty

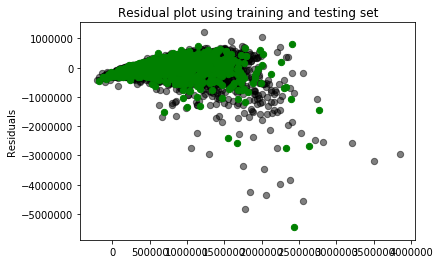
5 features/variables [PropertyID, UseCode, Transdate, censusblockgroup, ZoneCodeCounty] were not part of predicting sale price of houses.

**Building linear regression model Round 1:**

1. Handling missing values by replacing NaN by 0.
2. Used all the features/variables except the excluded ones, mentioned above.
3. Split the dataset into training 80% and testing 20%
4. Fitted the linear regression model
5. Calculated R^2 score, root mean squared error (RMSE), and mean absolute error (MAE) for the model.

|  |  |  |
| --- | --- | --- |
| **R squared** | **RMSE** | **MAE** |
| 68.13 | 251833.03 | 143728.49 |

**Residual Plot**

****

The R^2 score was decent. RMSE showed that the model predicted $251833 of the actual sale prices on testing set. MAE indicated that there were some predictions that were on target values for sale prices of houses, but there were some outliers. To visualize the differences, residual plot was plotted. From that its clear that there are quite several outliers which can be overestimating some predictive values by the model.

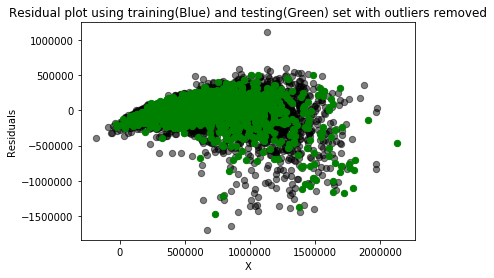
Thus, linear regression model was built again, but removing the outliers from the dataset.

**Building linear regression model Round 2:**

1. Z-score was calculated in order to remove the outliers by setting a threshold of z\_score > 5. This number was randomly picked after the z\_score array was generated.
2. Used all the features/variables except the excluded ones, mentioned above.
3. Dataset before outliers had dimensions of (11588,19) and after removing outliers, it was (11134,19).
4. Split the dataset into training 80% and testing 20%
5. Fitted the linear regression model again.
6. Calculated R^2 score, root mean squared error (RMSE), and mean absolute error (MAE) for the model.

|  |  |  |
| --- | --- | --- |
| **R squared** | **RMSE** | **MAE** |
| 78.04 | 173343.79 | 110559.56 |

**Residual Plot**:



Removing outliers showed a great deal of improvement in R^2 score by almost 10%, RMSE and MAE as well. RMSE and MAE were much smaller than when the outliers were not removed. The data points on the residual plot were more randomly scattered towards 0 which means the model had a good fit after removing the outliers.

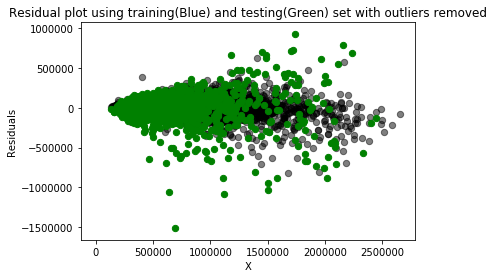
**Using Random Forest Regressor**:

1. The data without outliers was used for Random forest regression model
2. Training and testing set split was the same as for the linear regression
3. The metrics calculated were as follows:

R squared: 0.8421

Mean Absolute Error: 81537.9 degrees

**Residual plot:**



**Comparison of performance of two models:**

|  |  |  |
| --- | --- | --- |
| **Metrics** | **Linear Regression without Outliers** | **Random Forest without Outliers** |
| R squared | 78.04 | 84.21 |
| MAE | 110559.56 | 81537.9 |

Clearly, the random forest regressor did a pretty good job in predicting the sale prices of houses, compared to the linear regression model. Default parameters were chosen for random forest function using scikit-learn library.

**Finding Patterns in Features that Contributed Towards High and Low Prediction Accuracy**

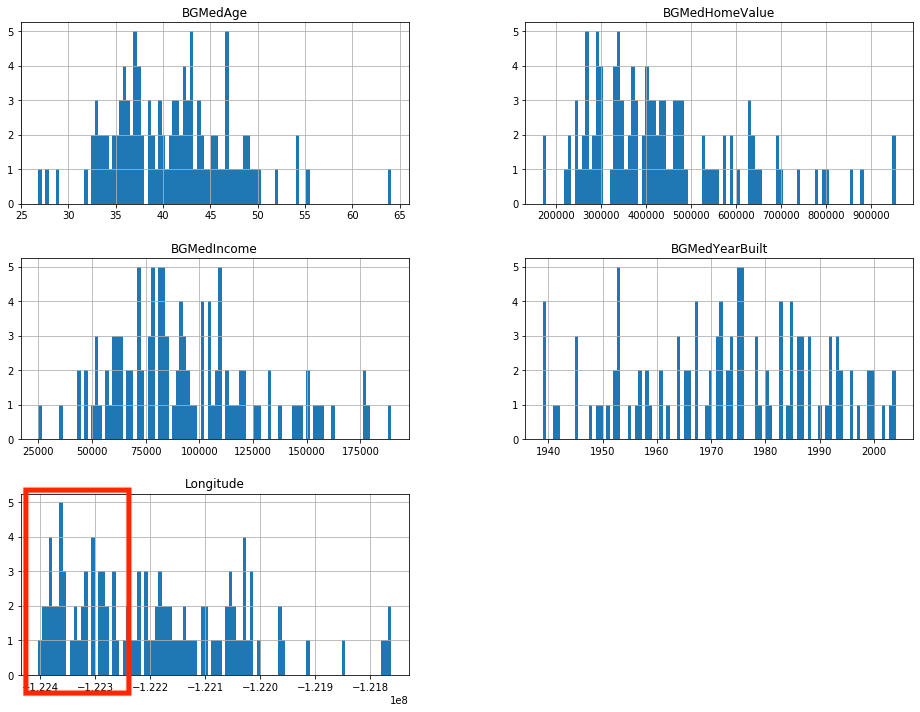
****

Fig A: For 95% percentile data

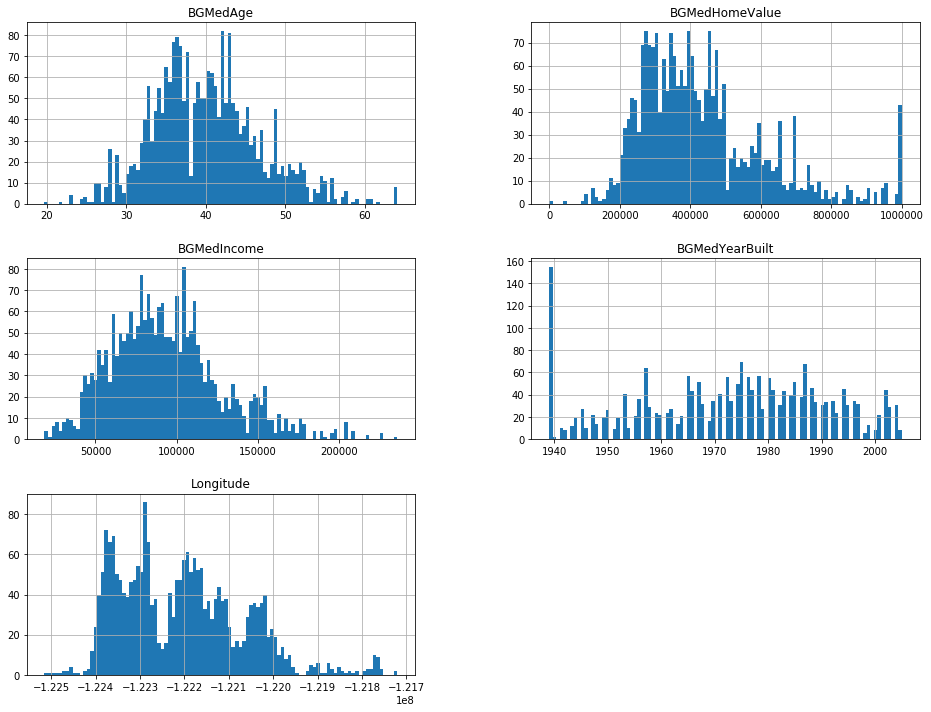
****

Fig B: For rest of the data

From the above comparisons on histograms (based on random forest predictions without outliers) to see if there are any patterns in the features involved for prediciton, the only feature noticeable was longitude (around -1224). Since there were not many strong patterns found, the next step is to perform some type of regularization technique on the regression model.

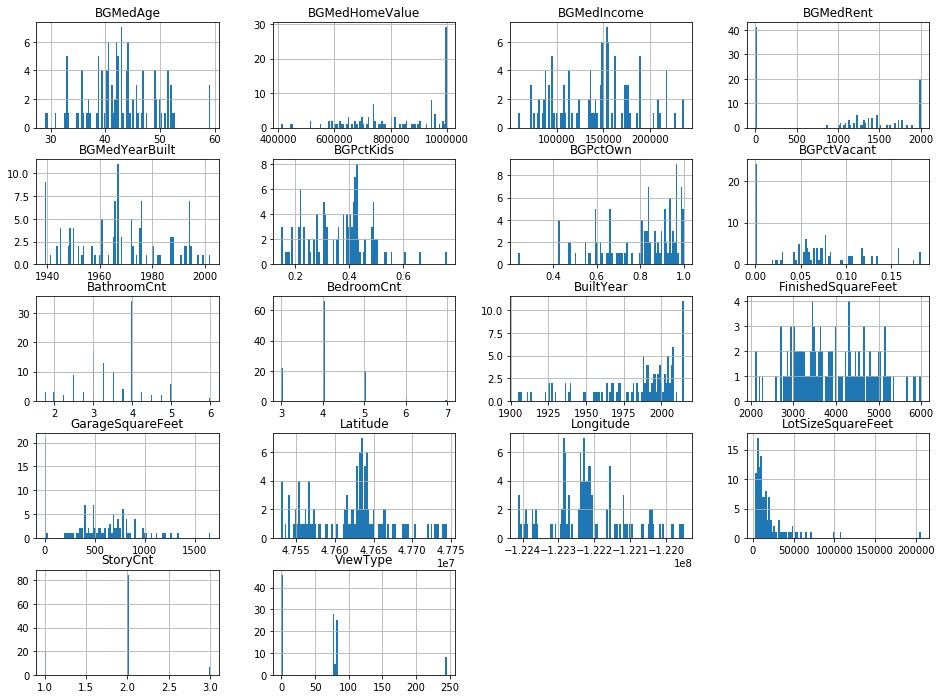
****

Fig C: After Ridge regression, for 95% percentile

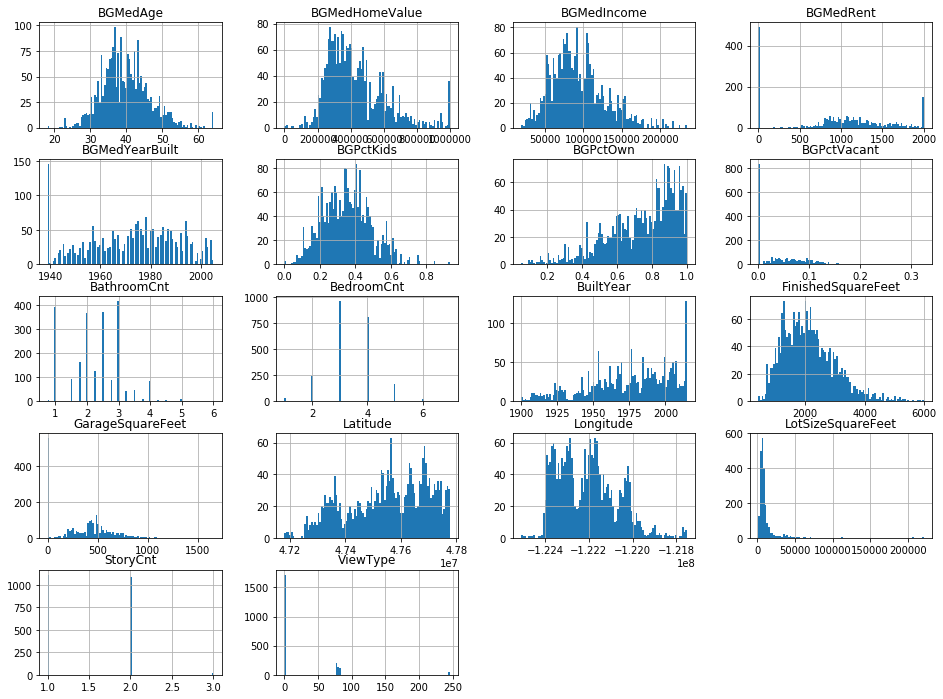
****

Fig D: After Ridge regression, for all the data

In the histograms above, we see some distribution and pattern for features like longitude, latitude, GarageSquareFeet, LotSizeSquareFeet, FinishedSquareFeet, BGMedRent that helps the model produce accurate predictions.

Conclusions:

1. Linear regression and random forest regression models were implemented where random forest regressor performed quite well with R^2 of 84.16 and residuals plot also showed random structure towards 0 for both axes (for random forest).
2. Without outliers, linear regression model performed much better compared to with outliers in the data
3. Ridge regularization on linear regression was performed as before that there were no patterns found in the features used for building the predictive model.
4. For the sake of comparison for finding patterns in the features, the predictions on testing set were extracted for those in 95% percentile ~ highest prediction error.
5. After ridge regression, few features like like longitude, latitude, GarageSquareFeet, LotSizeSquareFeet, FinishedSquareFeet, BGMedRent had patterns that contributed towards the accurate predictions of sale prices for houses.

**Python libraries used in the analysis:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error

from scipy import stats

from sklearn.ensemble import RandomForestRegressor

from sklearn.linear\_model import Ridge

All the references used in this analysis are mentioned in the code