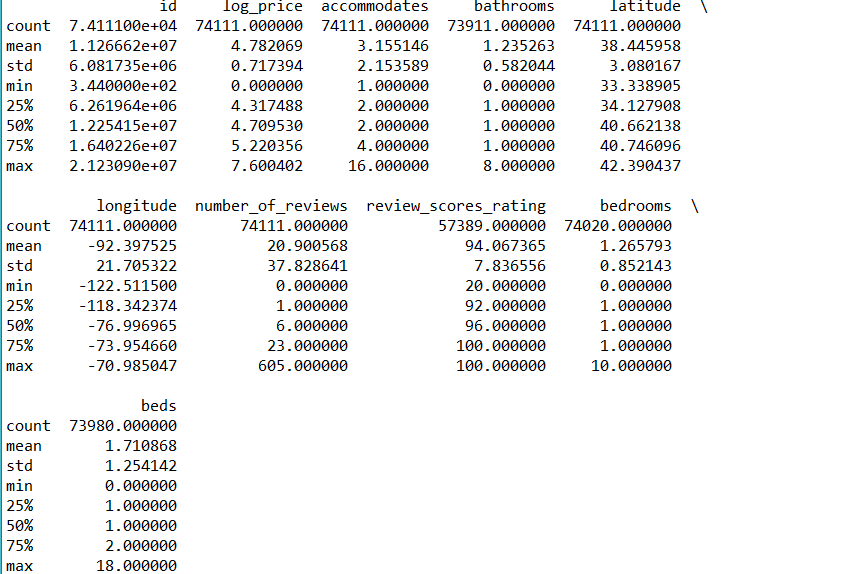
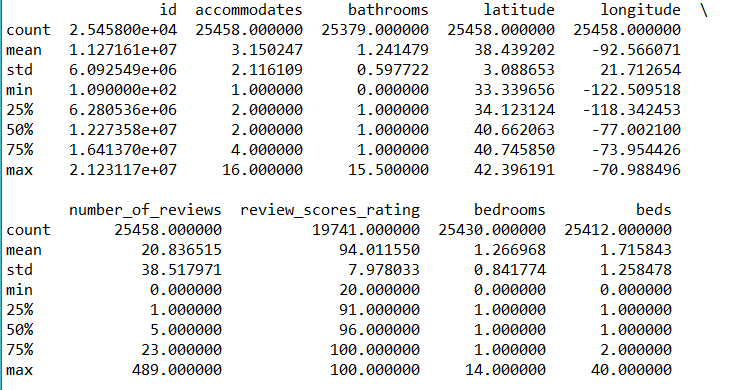
**AirBnB Price Prediction Challenge**

1. **Basic Stats on Train and Test data sets:**

**Train dataset:**

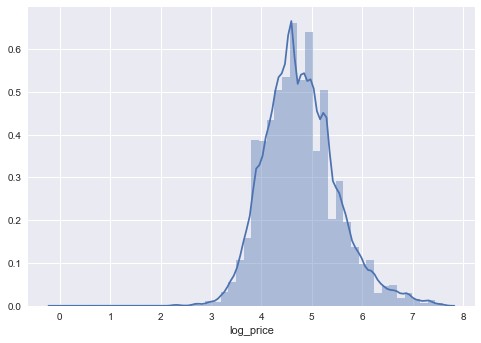


**Test dataset:**

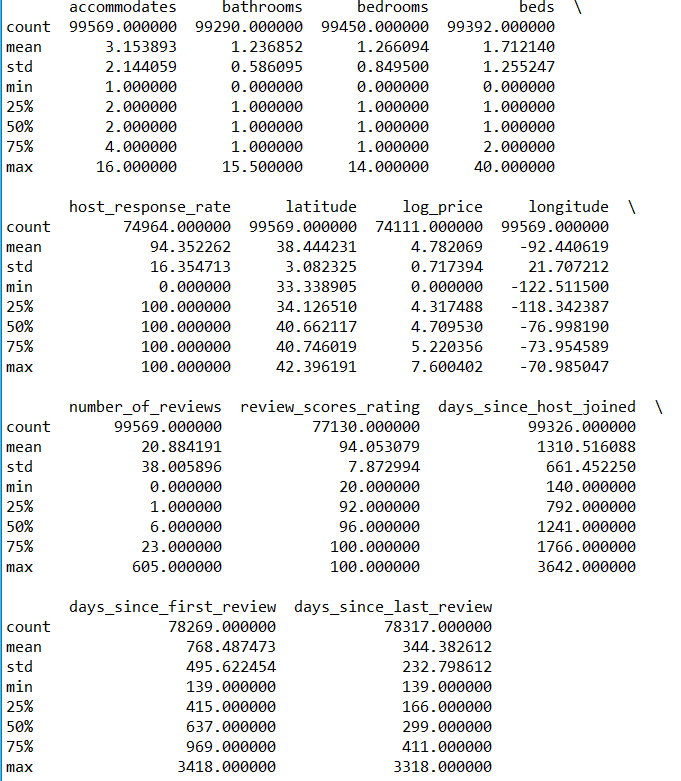


1. **Data exploration and pre-processing:**

Plotting univariate distributions: The log price is almost normally distributed with a mean of 4.782 and a standard deviation of 0.717



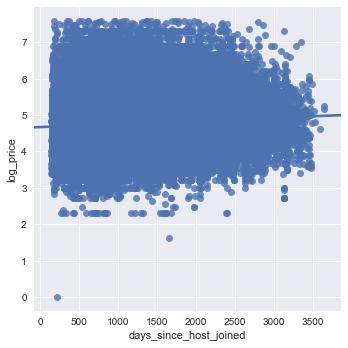
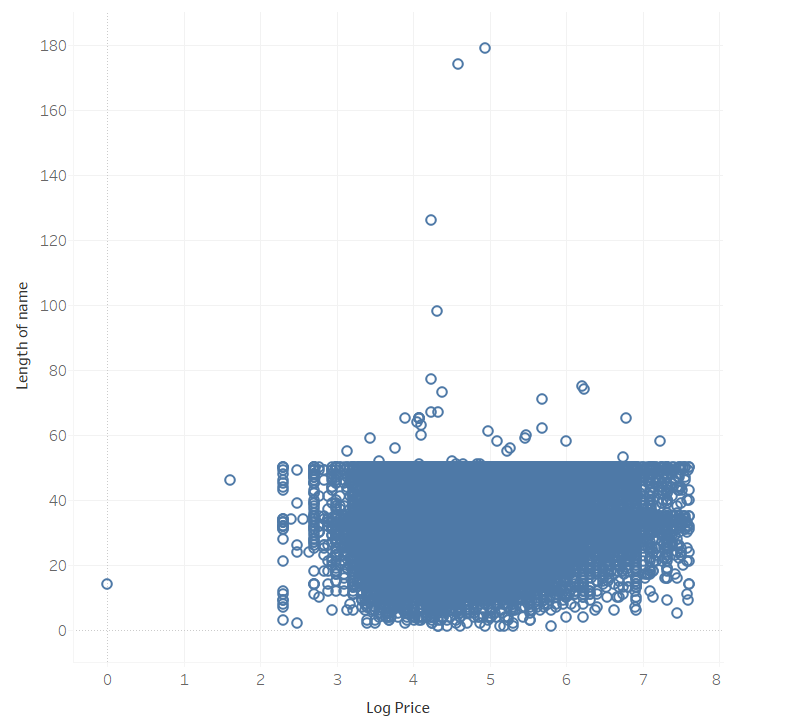
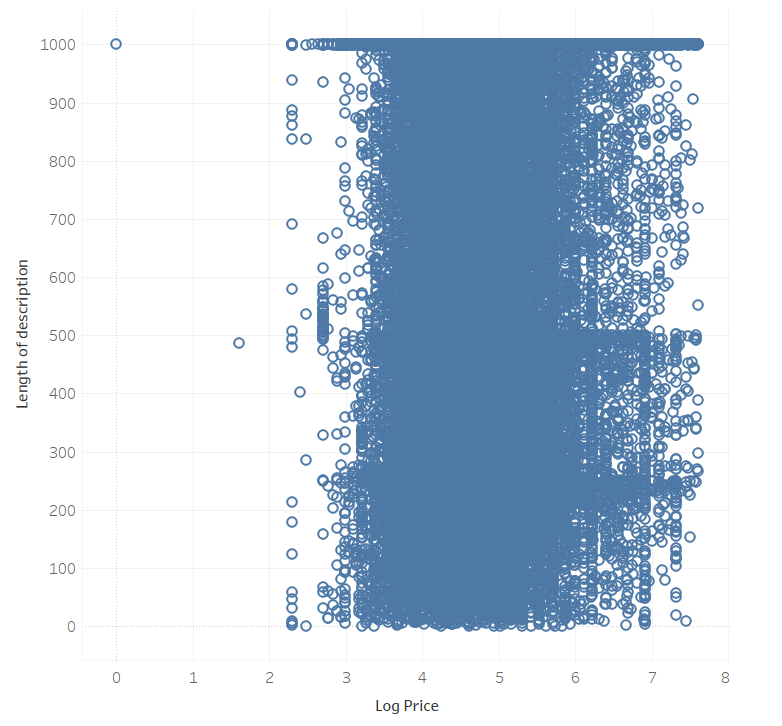
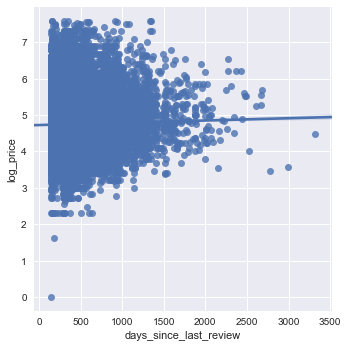
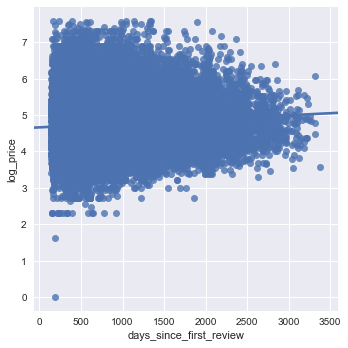
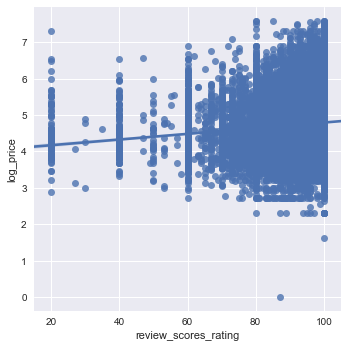
* Appending train and test datasets
* Converting host response rate to numeric, id to string, zipcode has '.0' towards the end for a few so remove that part
* Create additional fields : days\_since\_host\_joined, days\_since\_first\_review and days\_since\_last\_review using the date difference between the current day and these dates.
* Performed feature scaling on the dataset.
* Plotted a heatmap to observe the correlations between variable
* Checking for nulls or missing values on the combined(train and test) dataset:

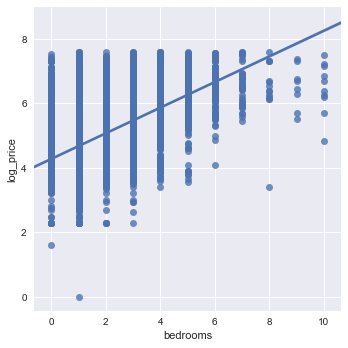
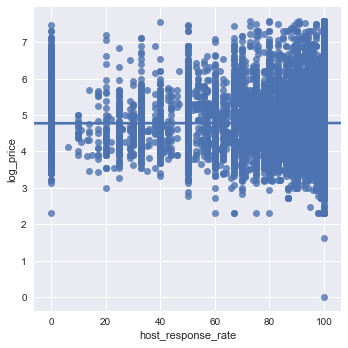
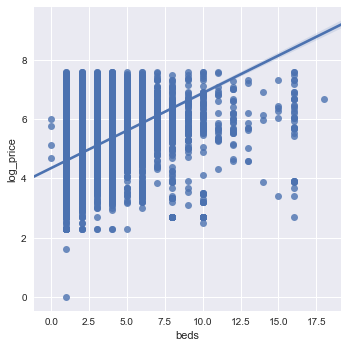
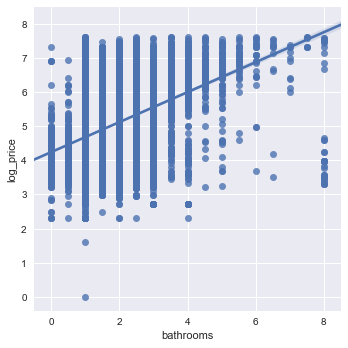


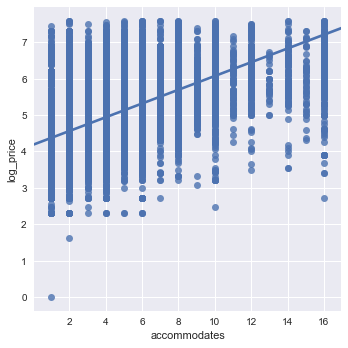
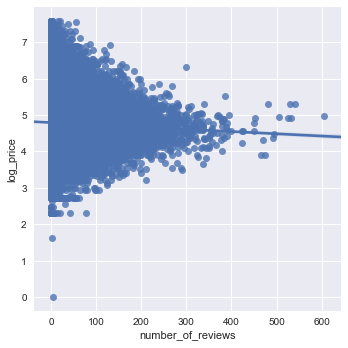
* Checked for missing values in the data –

Data imputation:

* + Obtained the missing zipcodes in data from the Python package uszipcode. Filled blanks in host\_response\_rate, review\_scores\_rating, days, with median
  + Filled blanks in beds as 1 as the Bedtype is shown as Real bed for all of them.
  + Filled blanks in bedrooms as 1 if the word 'bedroom' is present in the description else 0
  + Filled blanks in bathrooms with 0 if the word 'shared' is present in description otherwise same as number of bedrooms
* Plotted Bivariate distributions to see if there is any specific relation between the log price and numerical variables in train dataset.

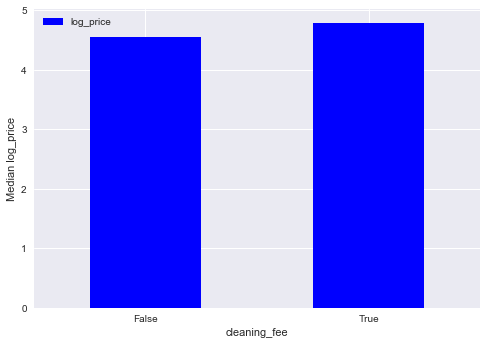
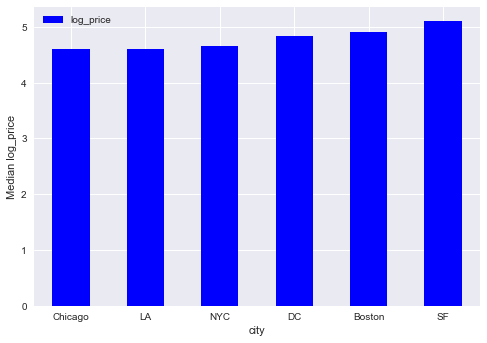
 

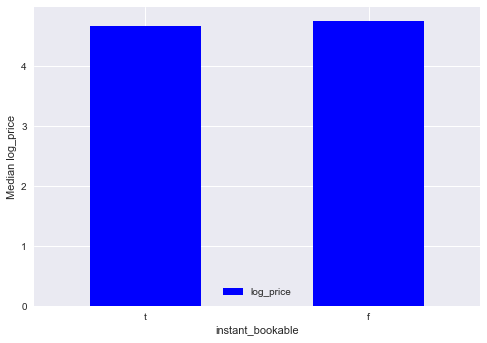
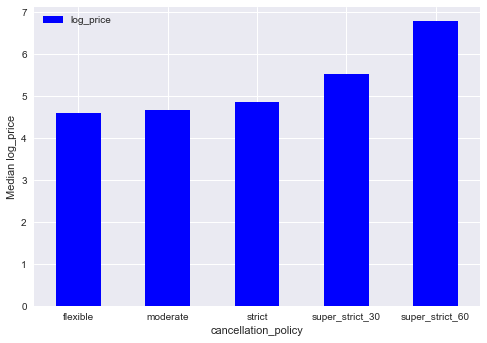


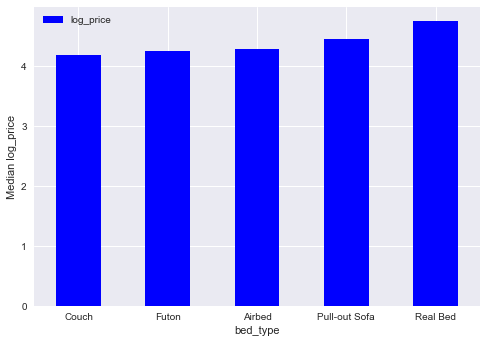
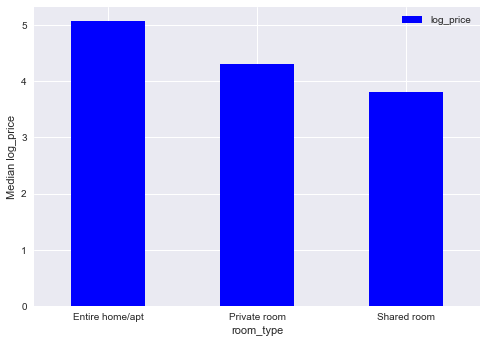
While the above plots do not convey a very strong correlation of the numerical features with target variable, we can obtain the pearson correlation value to check the strength of association of the numerical features with target variable.

|  |  |
| --- | --- |
|  | Value |
| accommodates vs log\_price | 0.567574 |
| bedrooms vs log\_price | 0.472422 |
| beds vs log\_price | 0.441953 |
| bathrooms vs log\_price | 0.355732 |
| review\_scores\_rating vs log\_price | 0.08418 |
| days\_since\_host\_joined vs log\_price | 0.078275 |
| days\_since\_first\_review vs log\_price | 0.056727 |
| longitude vs log\_price | -0.04753 |
| number\_of\_reviews vs log\_price | -0.03247 |
| days\_since\_last\_review vs log\_price | 0.009869 |
| latitude vs log\_price | -0.00219 |
| host\_response\_rate vs log\_price | 0.001423 |

Median price by Categorical variables:







* One-hot-encode categorical variables: ['property\_type','room\_type','bed\_type','cleaning\_fee','cancellation\_policy','city']

This step ensures that the categorical variables are converted to numerical variables.

* Performed Feature Scaling on all the numerical features on both train and test data sets.

1. **Model building:**
2. Trained a Random Forest Regressor model with cross-validation
3. Trained a XGBoost model with cross-validation
4. Ensemble model that combines Random Forest Regressor model and XGBoost model.

*Note: RMSE shown below is the mean of the k splits in k-fold cross validation.*

**Parameter Tuning within Random Forest Regression model:**

Case 1: Estimators=70

If latitude and longitude were removed in random forest regressor, RMSE: 0.43

Along with the above 2, if host\_response\_rate, review\_score\_rating are removed, RMSE: 0.43

Case 2: Estimators=100

Same case as above: RMSE: 0.43

Case 3: Estimators = 100

Considering all the numerical variables, RMSE: 0.39

***Observation:*** *Parameter tuning and feature selection has resulted in a decrease in error.*

**Parameter Tuning within XGBoost Model:**

Case 1:

(n\_estimators=100, learning\_rate=0.08, gamma=0, subsample=0.75,

colsample\_bytree=1, max\_depth=7)

RMSE: 0.39197

Case 2:

(n\_estimators=100, learning\_rate=0.1, gamma=0, subsample=0.75,

colsample\_bytree=1, max\_depth=7)

RMSE: 0.39039

Case 3:

(n\_estimators=500, learning\_rate=0.1, gamma=0, subsample=0.75,

colsample\_bytree=1, max\_depth=7)

RMSE: 0.38679

Case 4:

(n\_estimators=500, learning\_rate=0.1, gamma=0, subsample=0.75,

colsample\_bytree=1, max\_depth=6)

RMSE: 0.385995

Case 5:

(n\_estimators=500, learning\_rate=0.06, gamma=0, subsample=0.75,

colsample\_bytree=1, max\_depth=6)

RMSE: 0.3850

Case 6:

(n\_estimators=700, learning\_rate=0.06, gamma=0, subsample=0.75,

colsample\_bytree=1, max\_depth=6)

RMSE: 0.380456

***Observation:*** *Increasing the number of estimators has resulted in a decrease in error.*

***Ensemble model***

RMSE: 0.38366

**4. Final Results on the Test dataset for the chosen model:**

*Comparing the RMSE values of the above models, we can see that XGBoost model has a lower error as shown in the cross validation results on the train data set. Fitting this model to the test data, we have a*

RMSE: 0.37085

**5. Python Script:** Attached is the python script developed for this project in a word document.

