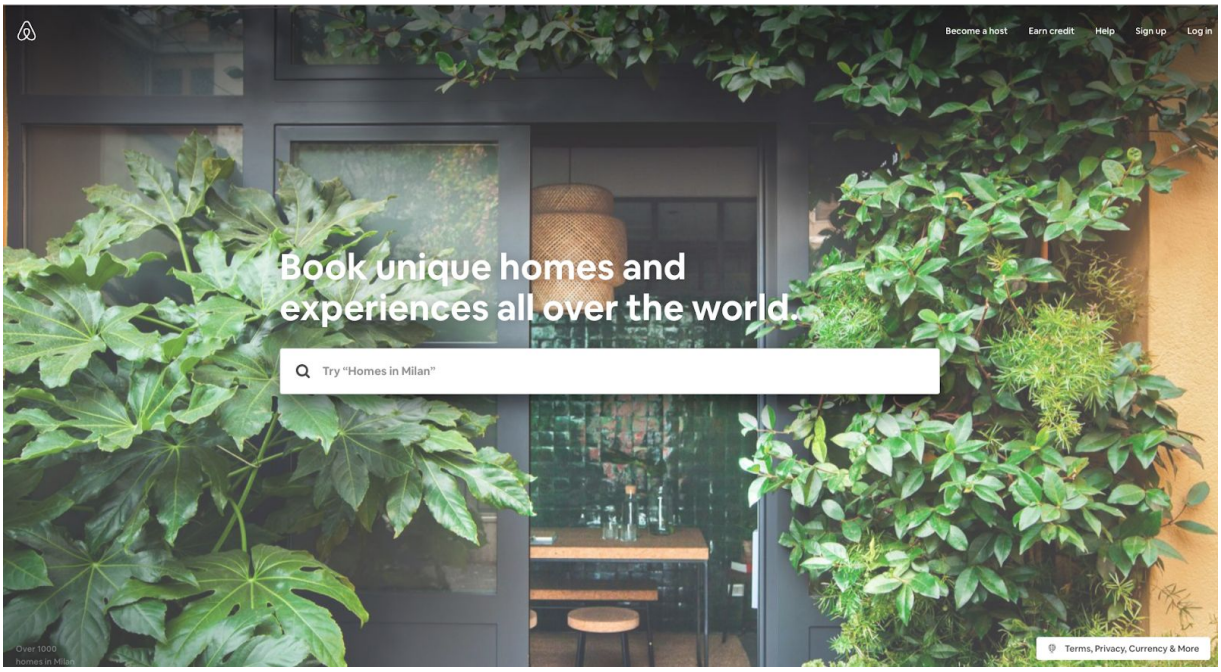
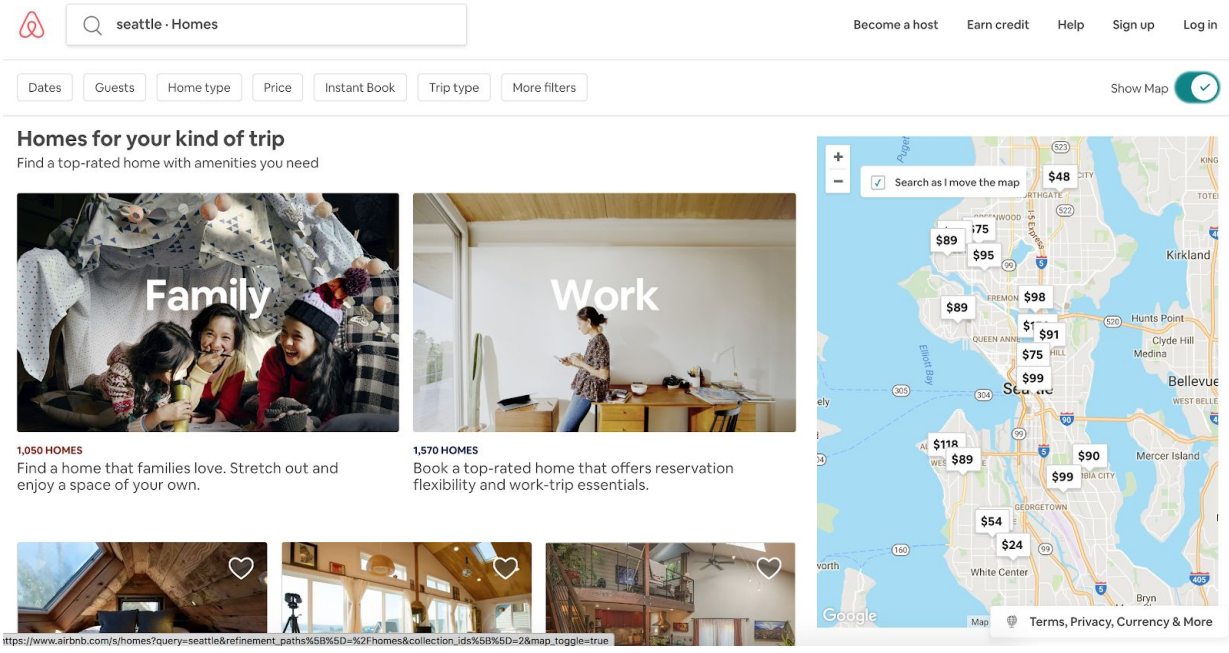
1st Capstone Project

Seattle airbnb Listing Price Prediction

Yueh-Tung (Nicole) Chao





Problem

Airbnb is an American company operating online marketplace and hospitality service for people to lease or rent short-term lodging. The company does not own any real estate. Instead, it is a broker which received percentage service fees in conjunction with every booking.

As from 2008, Airbnb has become one of the most popular hospitality services around the world, which has over 5 million lodging listings in 81,000 cities and 191 countries and has facilitated over 300 million check-ins.

One challenge that Airbnb hosts & Airbnb itself face is determining the optimal nightly rent price. As hosts, if charing too much, then the renter would select more affordable alternatives. On the other hand, if charging too less, then they lose the potential revenue. As for airbnb, if they can recommend optimal listing price to hosts to maximize their revenue, they can receive more service fees as well.

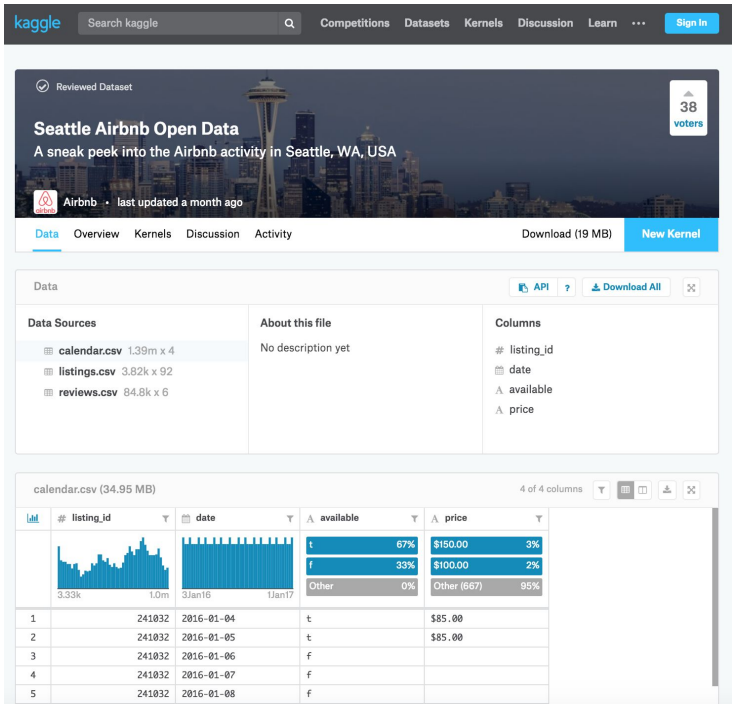
So in this capstone project, we propose to predict airbnb listing price so that both hosts & AIrbnb can optimize their revenue.

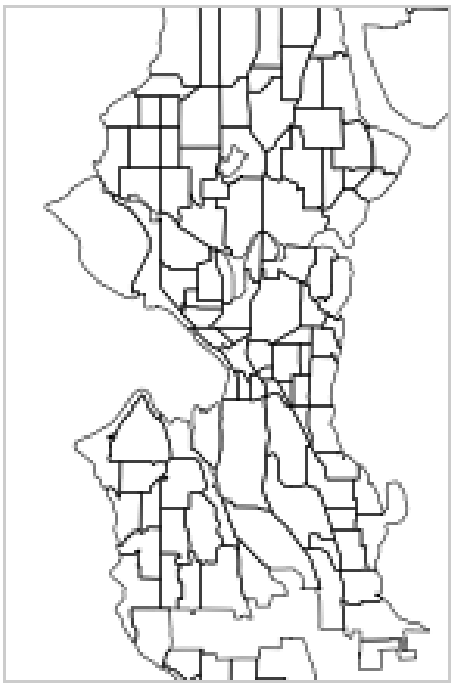
Client

As mentioned in previous session, by predicting listing price, it can be used for both hosts and airbnb for better pricing optimization & maximize the revenue.

Data Set

- The listing data of Seattle Airbnb can be found on Kaggle: Seattle Airbnb Open Data.
- To visualize data on map, the shapefile of Seattle neighborhood can be found on Zillow: Zillow Neighborhood Boundaries (See figures in next page).





Data Wrangling

First, we input data & do data wrangling. We conduct the following steps to clean up data and pick useful features.

1. Select columns that might be useful
    a. Visually inspect data & select columns that might be useful.
2. Analyze further & remove unwanted columns
    a. Originally there are 51 features/columns in the dataset, we further remove some unwanted columns.
    b. For example, we drop city-related columns as we are only predict airbnb listing price for one city now so these columns have same values.
3. Drop duplicates (if applicable)
    a. We also removed duplicate samples (rows) & duplicate columns (like neighborhood & neighborhood_cleansed).
4. Fix data types

    a. Then we look through all the data & fixed data types. For example, we identify categorical data & use sklearn LabelEncoder to encode then for further usage.
    b. We also combing latitude & longitude data using binning techniques.
    c. Replace some data with special characters, like '$' or '%' with numerical data.
    d. Replace dates with datetime object and transformed them into numerical data by calculating number of dates between the dates and the last-scraped-date, etc.
5. Fix missing values
    a. Plot scatter plot for columns with missing values and inspect the trend. Confirmed that most of the features are either linear or random to target.
    b. Fill missing values with mean.
6. Find & fix outliers
    a. Remove minimum night outlier by cap it to be 99.5 percentile of the data.
7. Drop unnecessary features as we only have 3818 data points so we might not want to have over 20 features.
    a. Plot scatter plot of each feature vs. target, i,e, price and drop some unnecessary columns, i.e., columns that do not seem to affect price.

Initial Findings

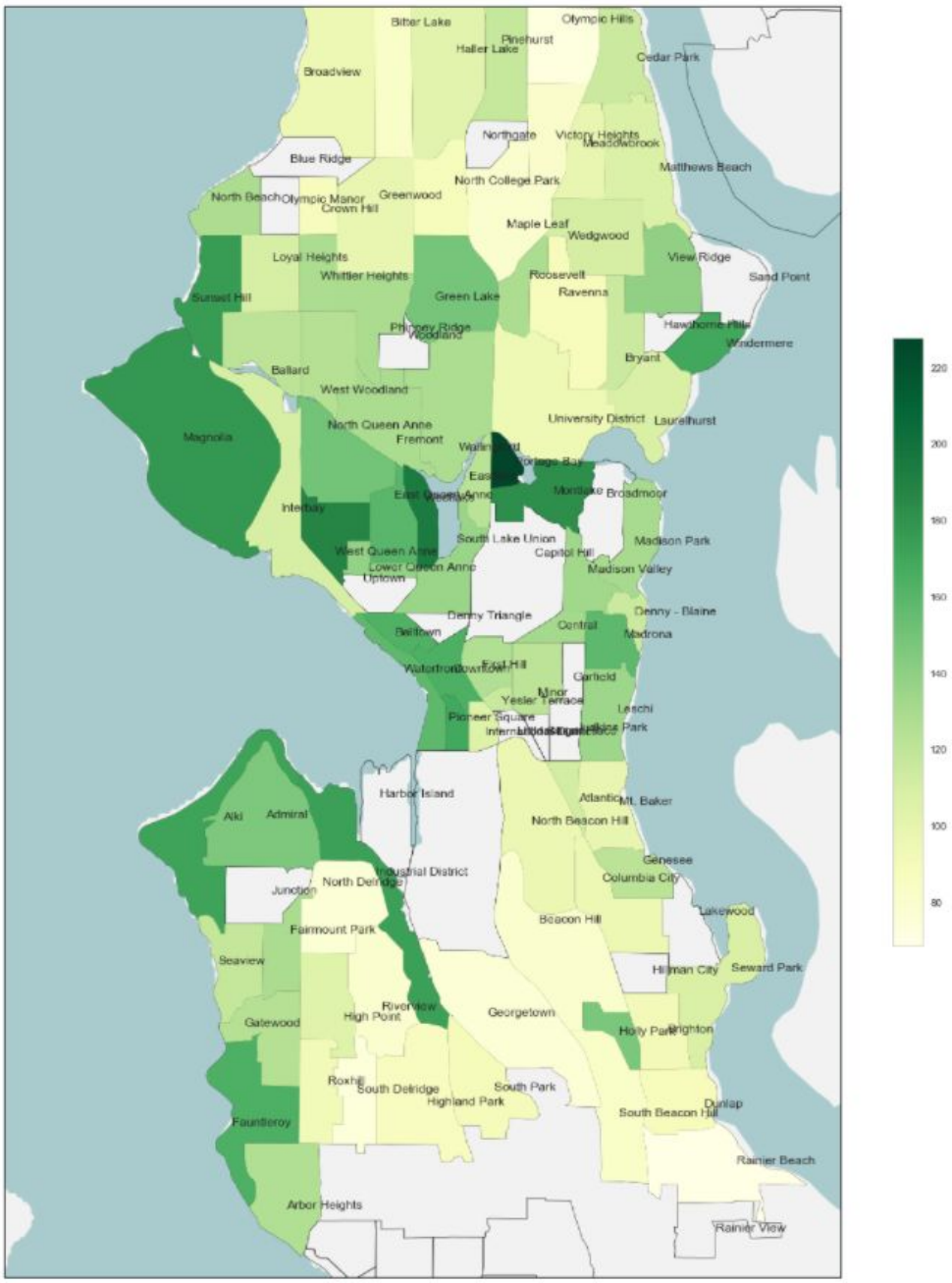
During exploratory data analysis, we ask the following questions:

1. What might be the most important features that affect listing price?
2. Does location/neighborhood affect listing price?
3. If yes, what made these neighborhood special?

and ask follow-up questions if needed.

Initial findings are:

1. As we expected, listing price is correlated to accommodates, bedrooms, bathroom, neighborhood, and property type.
    a. Interestingly looks like neighborhoods facing water (either bay or lake) seem to have higher listing prices, like Portage Bay (See the figure in next page).



b. However there might be some exceptions:
    i. There might be extremely high listing price although there's only one bedroom, but rarely happened.
    ii. If the property has a lot more bathrooms, ex. 8, but it's a dorm, the listing price would not increase proportionally.

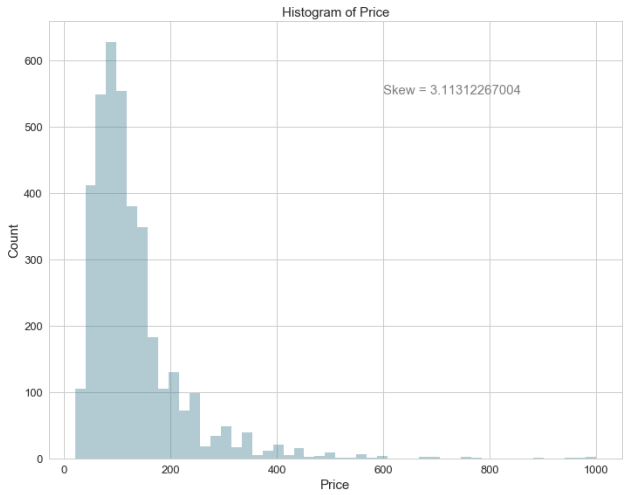
iii. If the property has a lot more bathrooms, ex. 8, but can only accommodate 2 people, the listing price would not increase proportionally, either.
iv. If the property is a big house which can accommodate 14 people but its accommodates vs. bedrooms ratio and/or accommodates vs. bathrooms ratio are higher, that means more traveler need to share the bedrooms / bathrooms, the lisilistingng price would not increase proportionally, either.

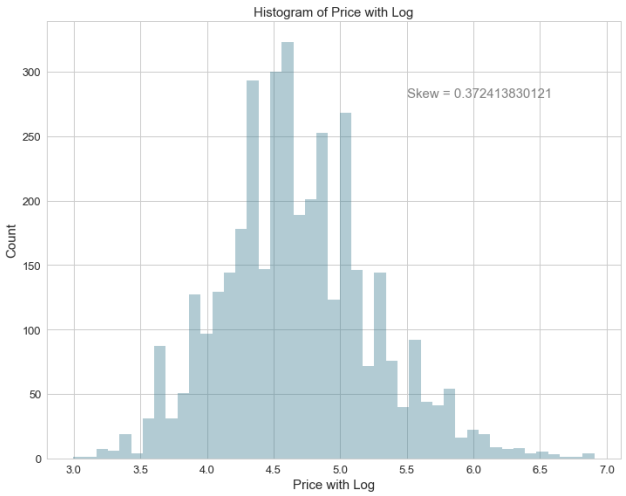
2. Property-type-wise, most of the property type of Seattle listings are house and apartment.
    a. Wallingford & First Hill have most number of listings whose property type is house.
    b. First Hill & Belltown have most number of listings whose property type is apartment.
    c. Over 13% of Seattle Airbnb listings are located in First Hill.
3. In terms of bedrooms and bathrooms, 1 bedroom 1 bathroom Airbnbs which can accommodate 1 ~ 4 people are most common in Seattle.
    a. Over 50% of Seattle Airbnb listing are of this type.
    b. About 10% of them are located in First Hill.
    c. For the rest of Airbnbs, not many of them can accommodate more than 10 people. In fact, only 1.3% of listings can accommodate 10 or more than 10 people, and other than First Hill, most of them are located in some suburban area.

Modeling

With insights based on exploratory data analysis (EDA), we start to train predictive models.

We check the distribution of our target, which is price & confirmed that taking log can make it distribute more normally & skew is much improved.





We first try the following five models:

1. K-Nearest Neighbors
2. Linear Regression
3. Ridge Regression
4. Lasso Regression
5. Random Forest

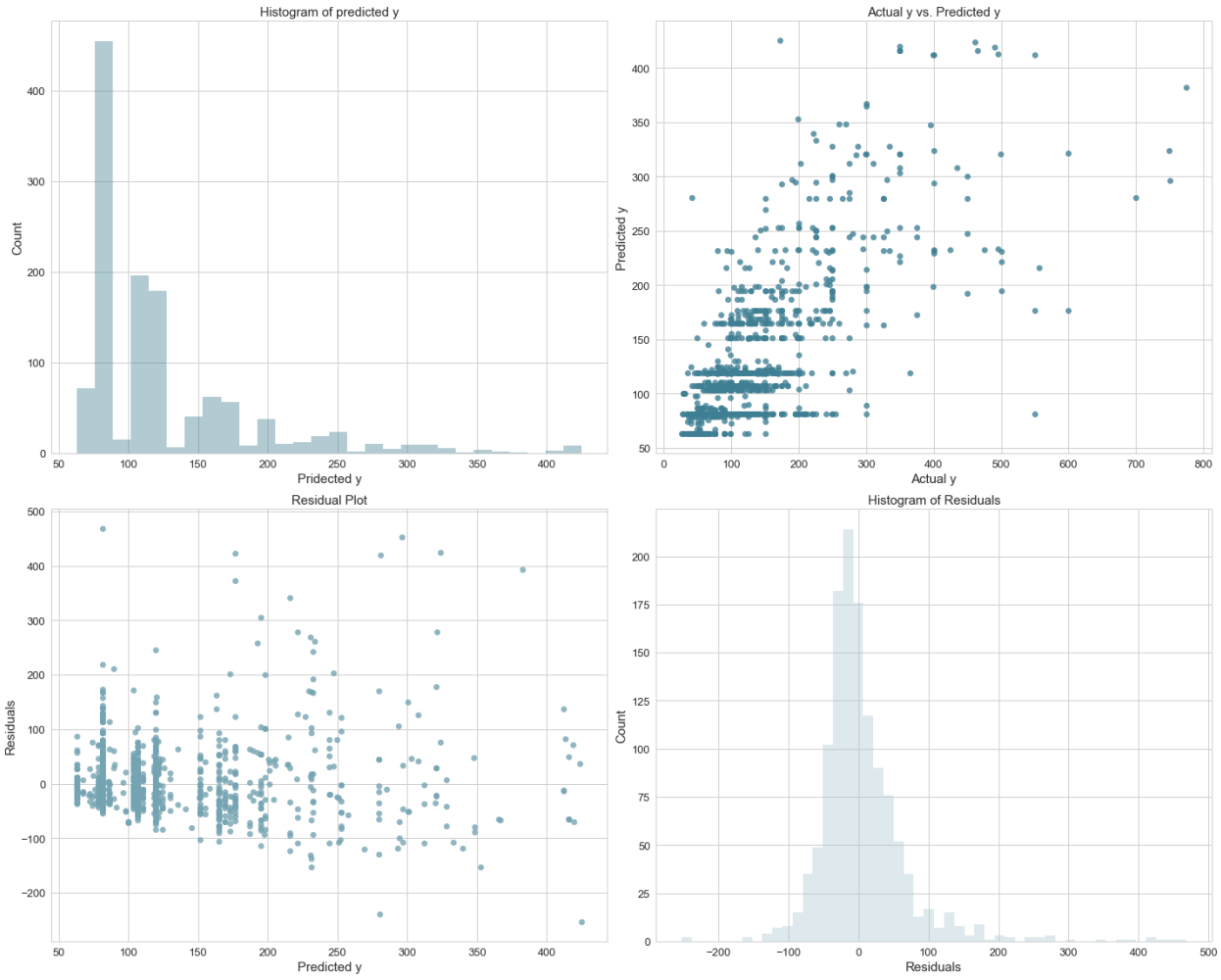
With three features:

1. Accommodates
2. Bathrooms
3. Bedrooms

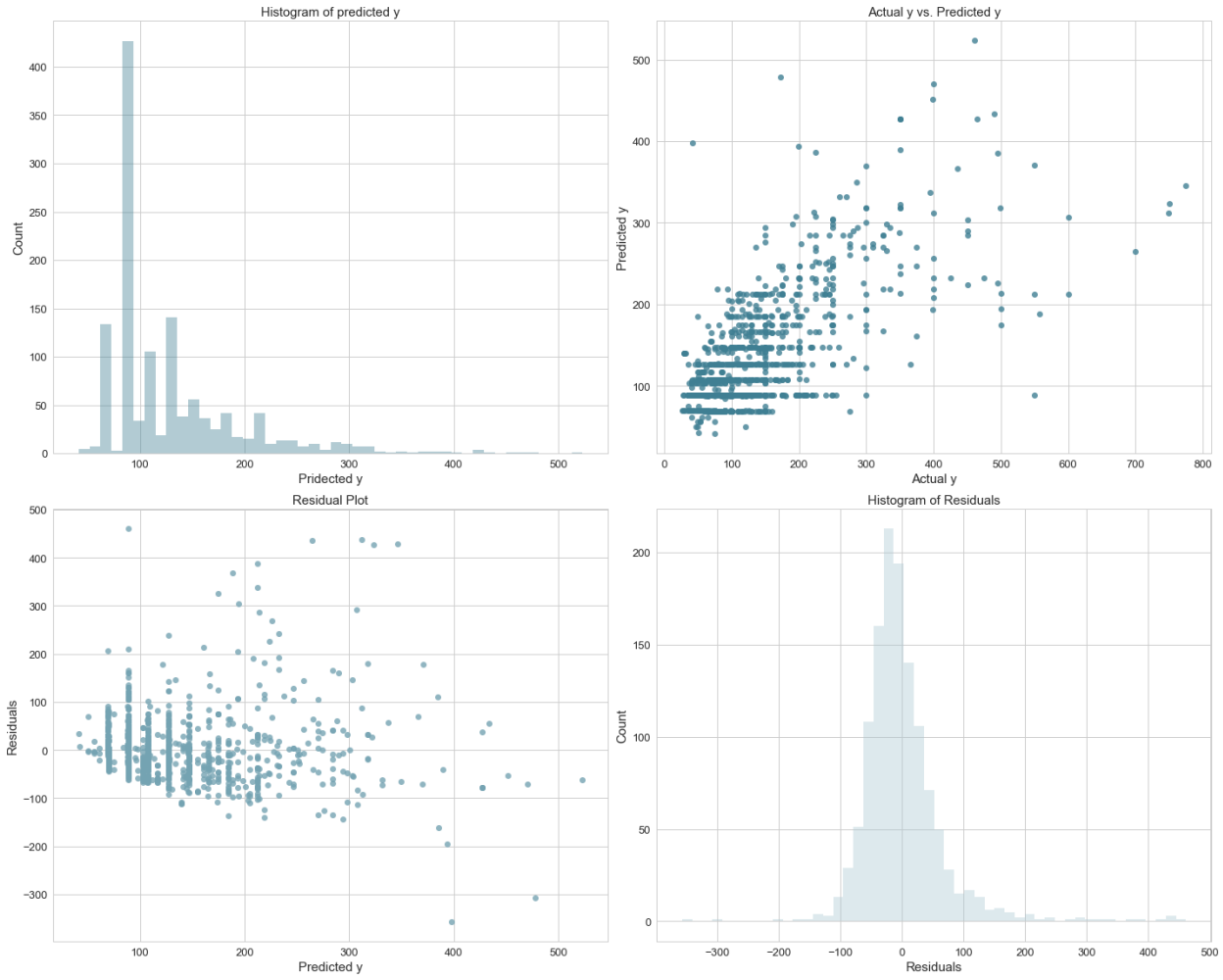
Test size of Train-test-split is set to 33%.

For linear models including Linear Regression, Ridge Regression & Lasso Regression, we also try to take log on price:

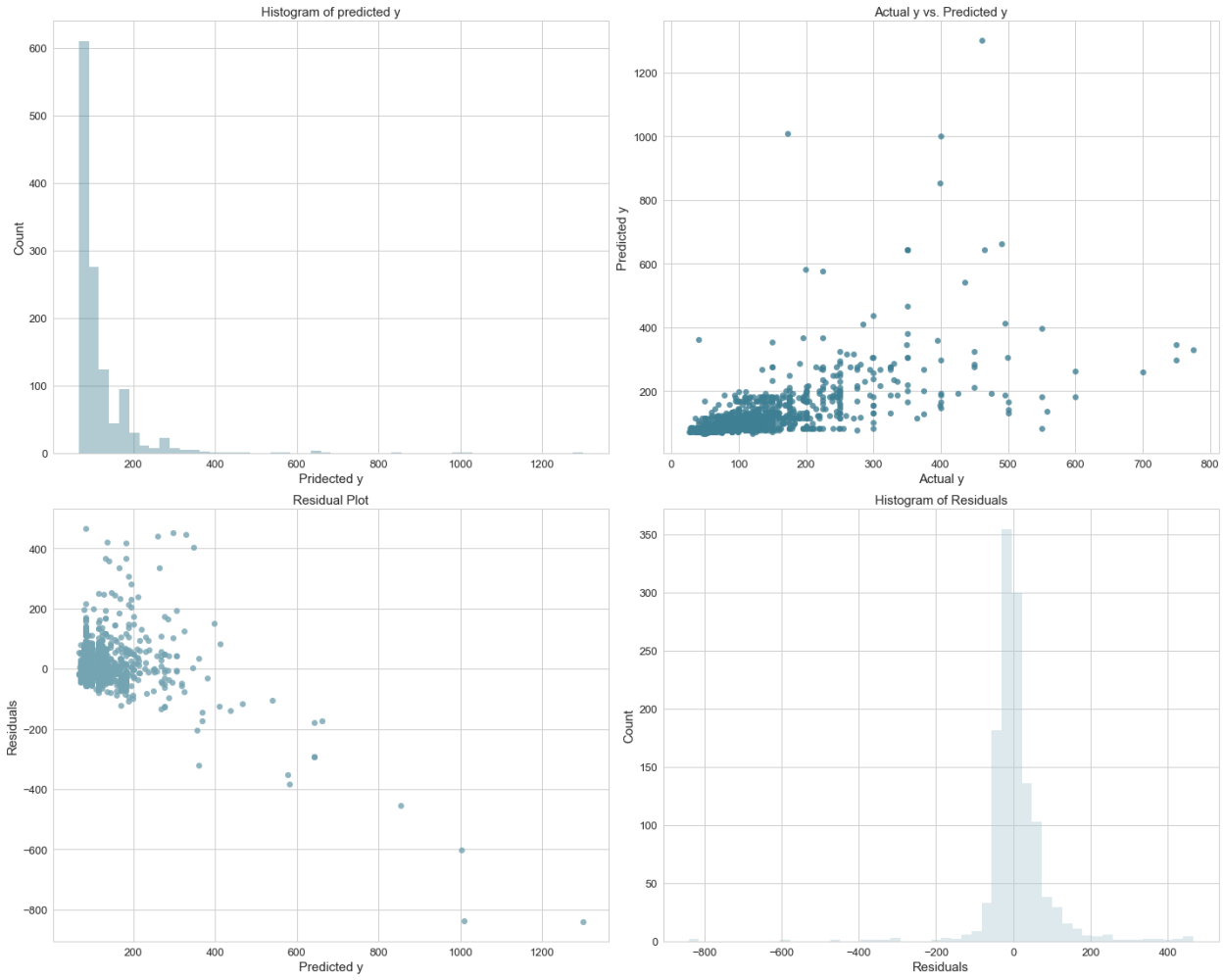
1. K-Nearest Neighbors
    a. Best hyperparameter for n_neighbors is 33. RMSE is 63.8775604886.



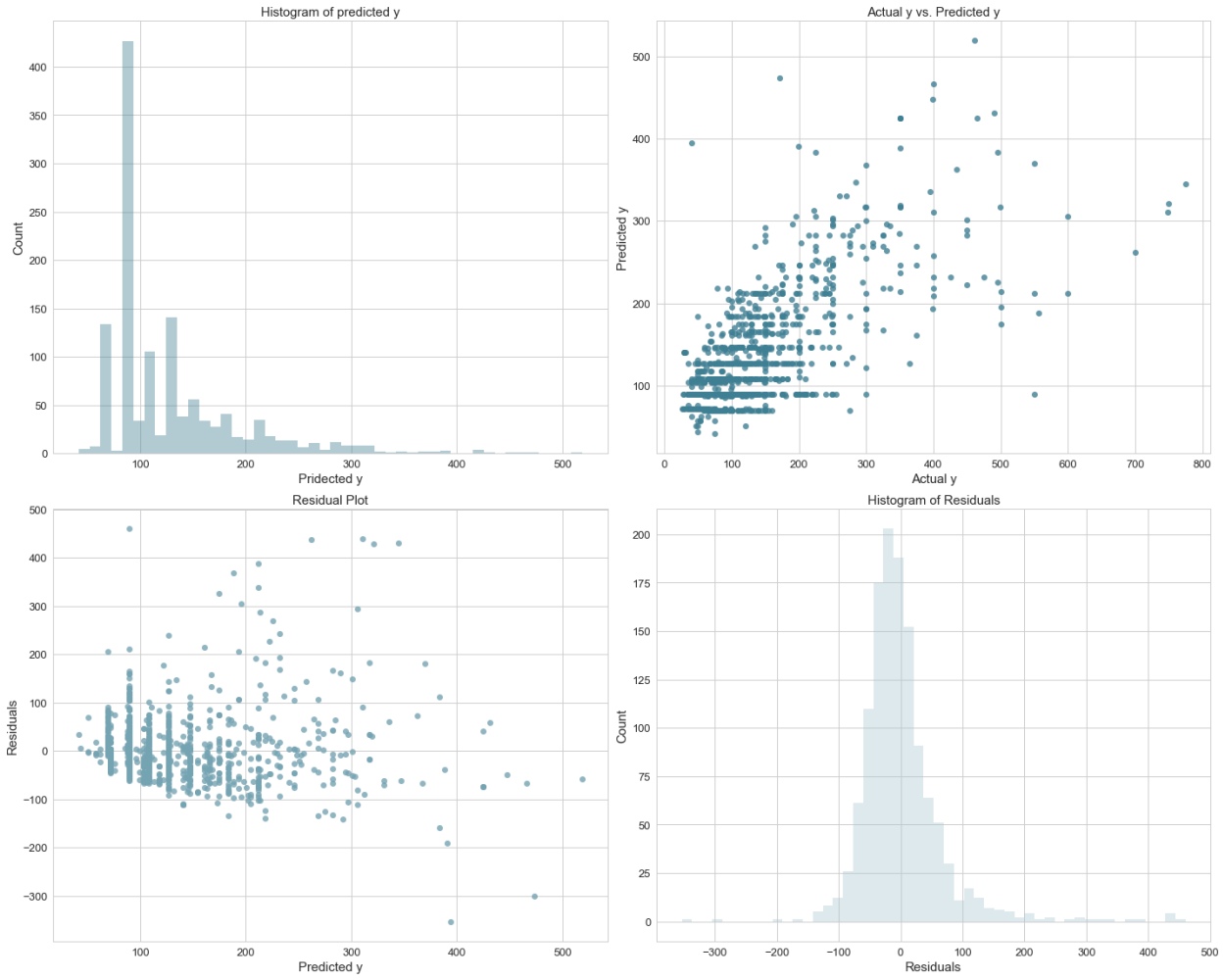
2. Linear Regression
    a. If not taking log on price: RMSE is 65.782884005.



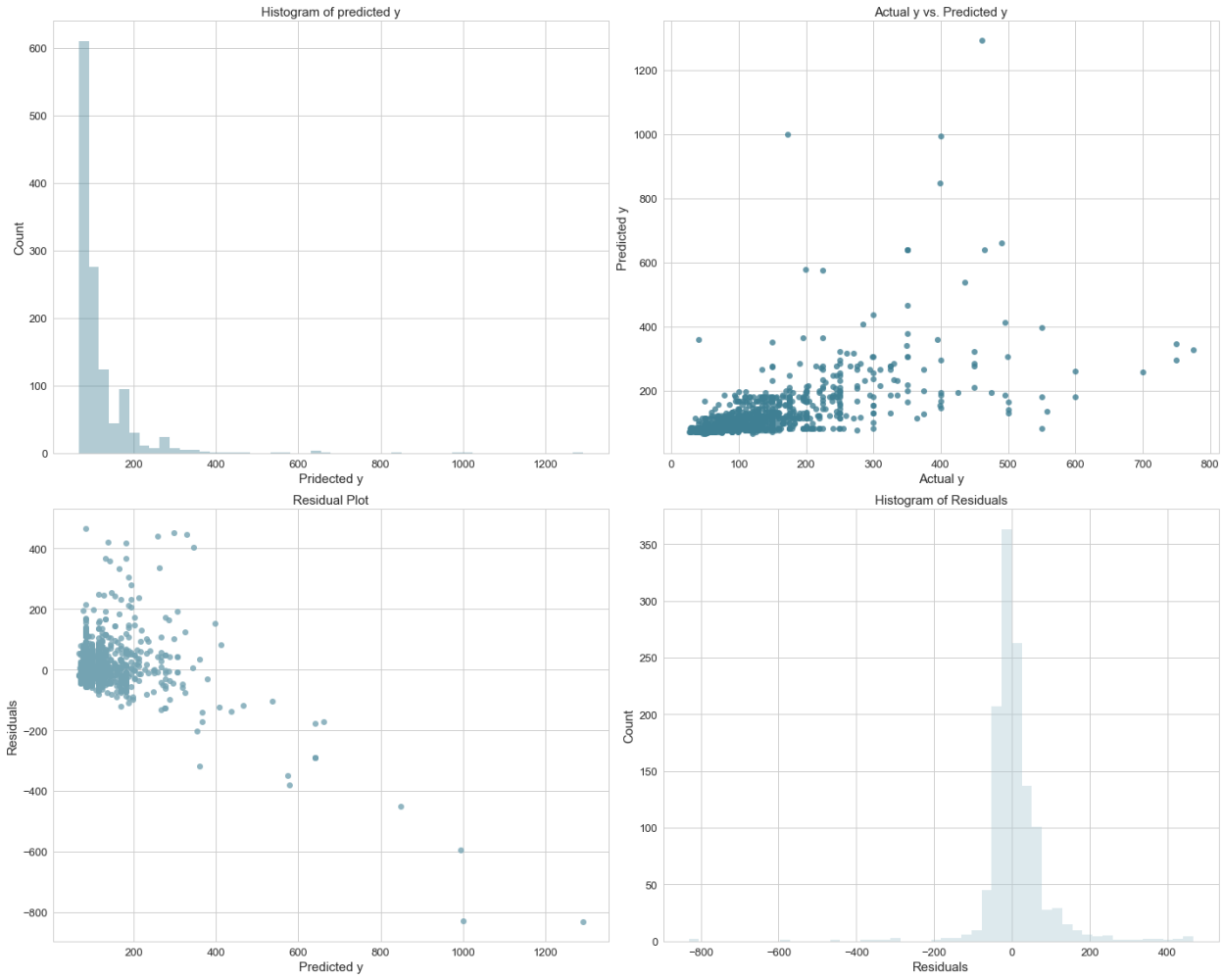
b. If taking log on price: RMSE is 80.6990729468.



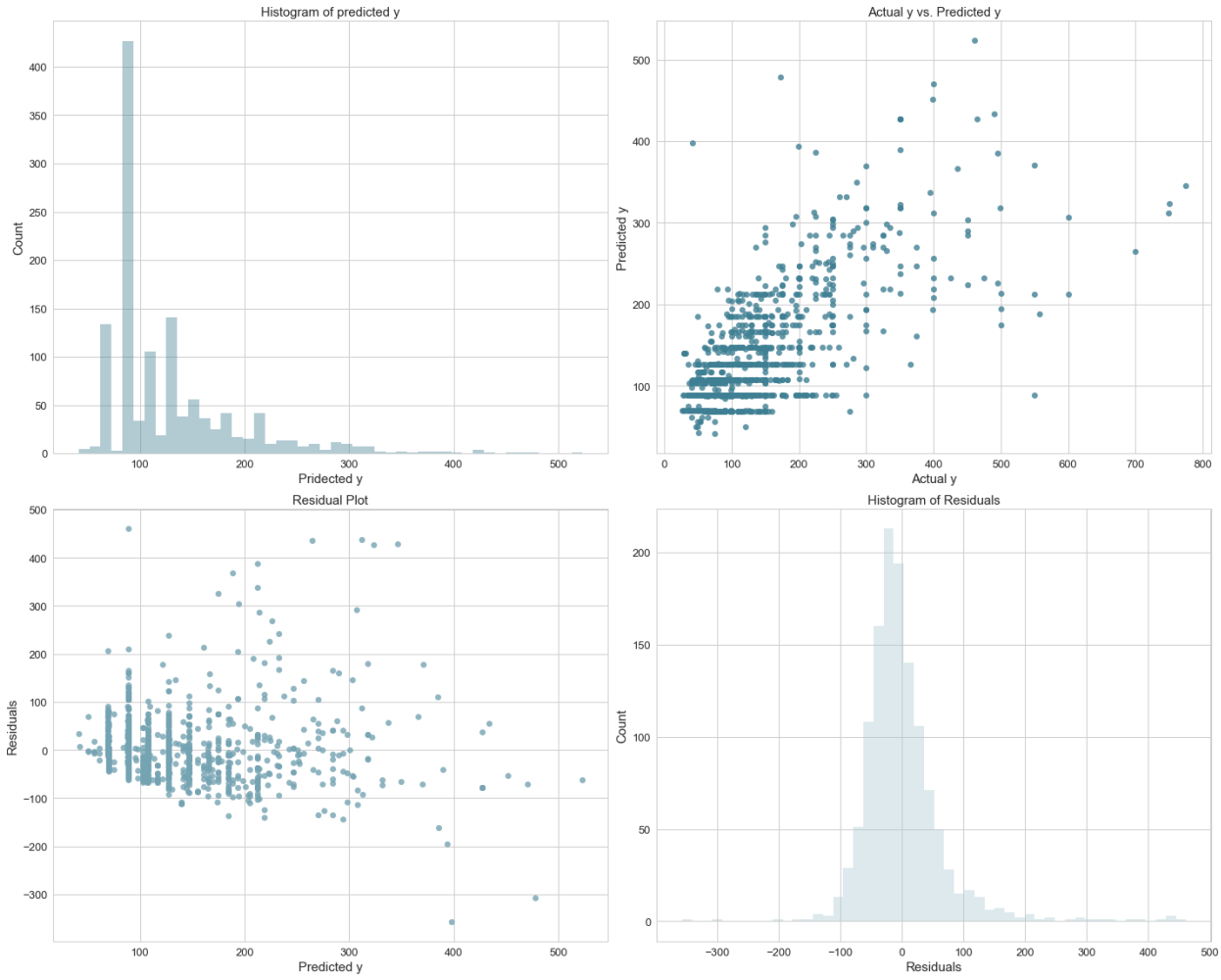
3. Ridge Regression
    a. If not taking log on price: Best hyperparameter for alpha is 50. RMSE is 65.7372413634.



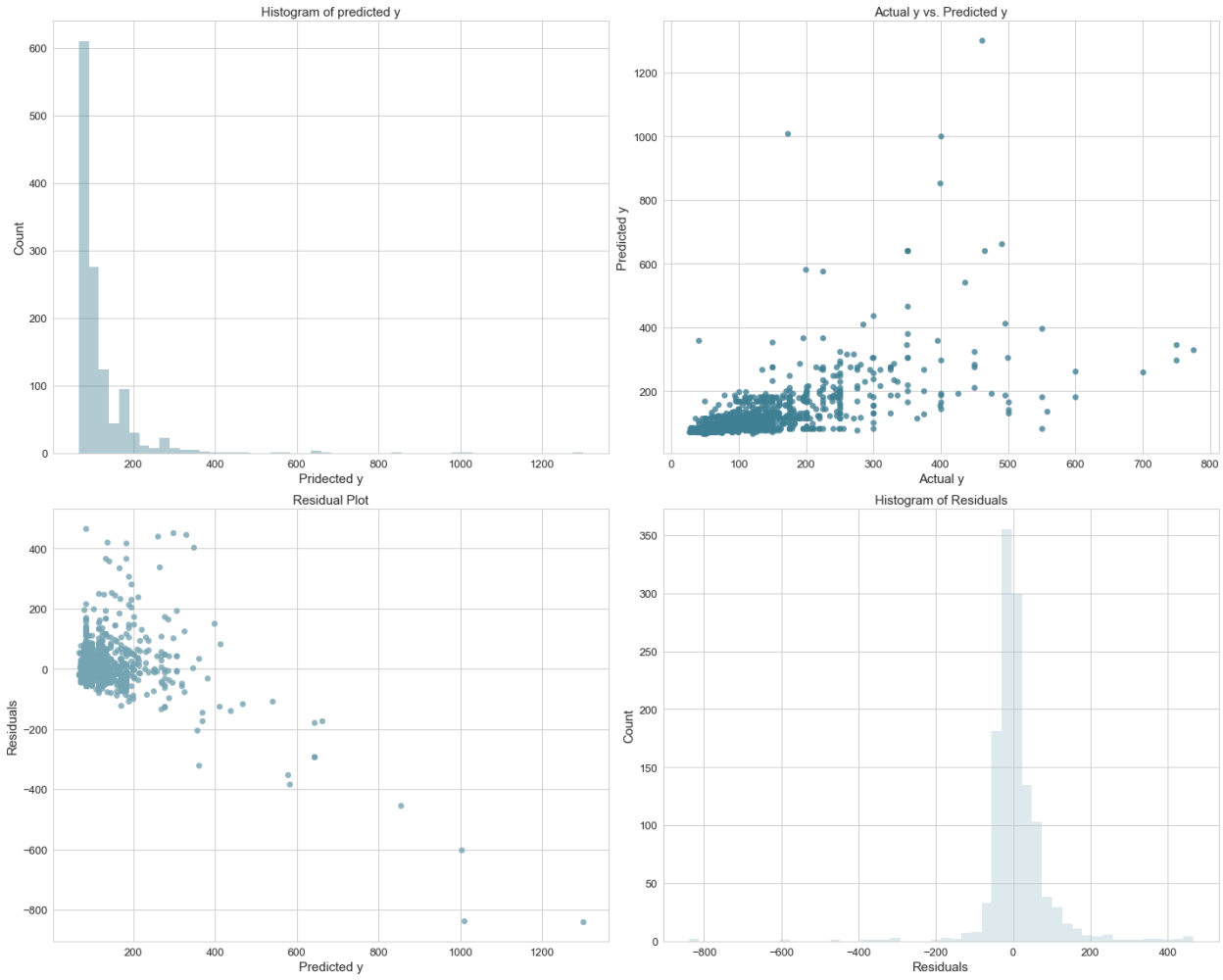
b. If taking log on price: Best hyperparameter for alpha is 10. RMSE is 80.4074239407.



4. Lasso Regression
    a. If not taking log on price: Best hyperparameter for alpha is 0.0001. RMSE is 65.782902475.

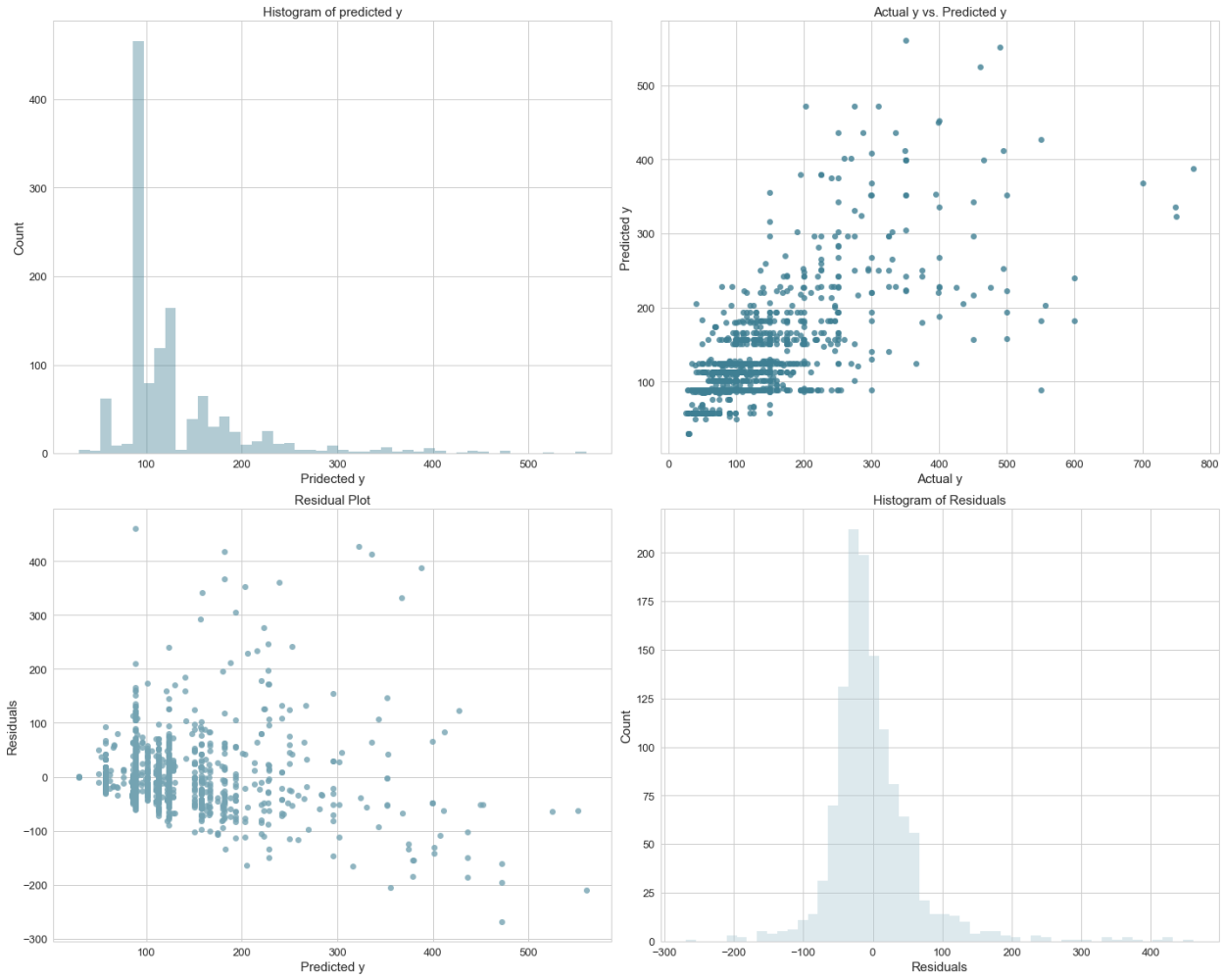


b. If taking log on price: Best hyperparameter for alpha is 0.0001. RMSE is 80.6875515723.



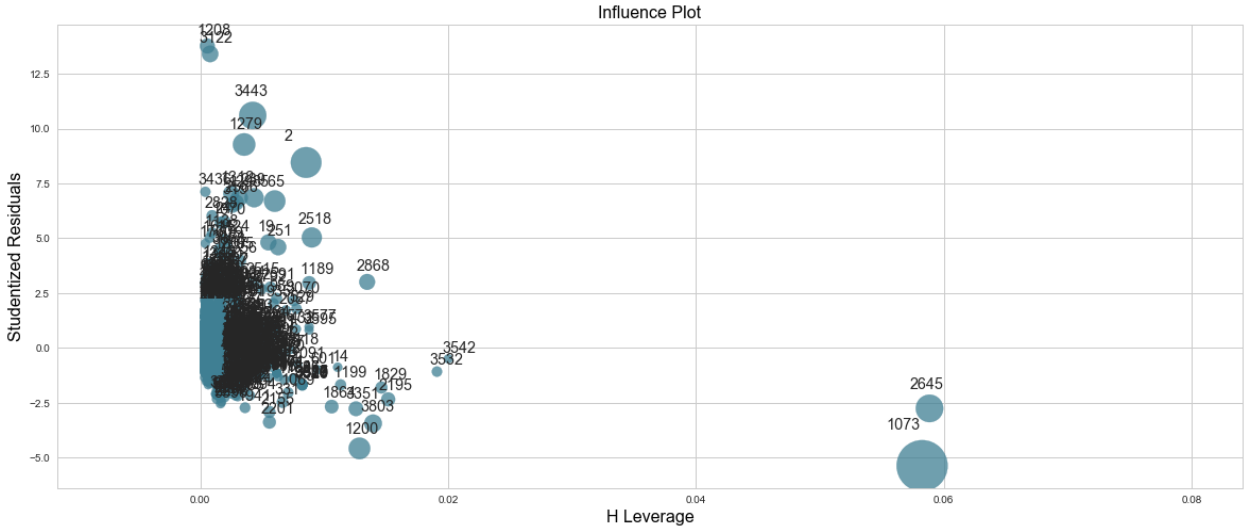
5. Random Forest

    a. Best hyperparameter for max_depth is 10. Best n_estimators is 30. RMSE is 65.8094760545.



The first trial shows that

1. KNN has best Root-Mean-Squared-Error (RMSE).
2. Although taking log on price did make it distribute more normally, RMSE isn't better.
    a. We plot influence plot to check if there's high leverage point:



b. And by removing high leverage points, we trained the model again. However RMSE still isn't better on linear models.
c. By analyzing further, it looks like there are less data points with price above about 300 and looks like they have a different linear relationship. For this **piecewise linear regression** is sometimes used.

We then get back to the models. As KNN & Random Forest have best results so far, we analyze outliers of these two models further.

For both models, there are many data with **actual price greater than 600 while predicted price is less than 400**. By looking into them, we found that over half of them are **house** in terms of **property type**, which means it worths to try to add property type into our features.

We then try to add property type into our features:

1. K-Nearest Neighbors
    a. Best hyperparameter for n_neighbors is 17. RMSE is 66.2413412525.
2. Random Forest
    a. Best hyperparameter for max_depth is 10. Best n_estimators is 80. RMSE is 65.4463938358.

Without property type, KNN has RMSE 65.7802492126, random forest has RMSE 66.1536873354.

After adding property type, KNN gives 66.2413412525, random forest gives 65.4463938358. **Looks like KNN result degraded while Random Forest is improved.**

Also in exploratory data analysis (EDA), we also found that price is also affected by neighborhood. We further added into our features.

1. K-Nearest Neighbors
    a. Best hyperparameter for n_neighbors is 4. RMSE is 73.3585695725.
2. Random Forest
    a. Best hyperparameter for max_depth is 10. Best n_estimators is 90. RMSE is 63.5181692736.

By adding neighborhood, we can further improve Random Forest results a little bit. RMSE is improved from 65.4463938358 to 63.5181692736.

However KNN results degrade a lot. By inspecting the KNN models, number of 'n_neighbors' became smaller & smaller when we add more features.

Note that KNN depends on similar neighbor data points to get better prediction results. It looks like when we add more features, it increased dementinality. W/ the curse of dimensionality. Number of similar data points seems becoming less, so RMSE started to get higher.

On the other hand, random forest by nature automatically select useful features for splitting so did not have this issue.

Last but not least, we try the Gradient Boosting model.

Initial trial with hyperparameter tuning only on n_estimators shows that RMSE is 63.749581598.

If we tune more hyperparameters, like learning_rate and max_depth, RMSE is further improved to 63.1581040609, which is the best out of all the models.

In my opinion it proved that with the techniques of gradient boosting, it can indeed improve prediction accuracy.

Conclusion

1. We tried 6 different models on Airbnb listing price prediction
    a. K-Nearest Neighbors
    b. Linear Regression
    c. Ridge Regression
    d. Lasso Regression
    e. Random Forest
    f. Gradient Boosting
2. KNN & random forest outperforms linear regression models. After adding more features, random forest performs better than KNN.
    a. If looking at KNN results, after adding more features, best parameter for n_neighbors became less. It might be because KNN depends on similar neighbor data points to get better prediction results. When adding more features, it increased dementinality. W/ the curse of dimensionality. Number of similar data points seems becoming less, so RMSE started to get higher.
    b. On the other hand, random forest by nature automatically select useful features when splitting so did not have this issue.
3. Linear regression models did not perform well because here are less data points with price above about 300 and looks like they have a different linear relationship. For this piecewise linear regression is sometimes used.
4. Gradient boosting model gives the BEST RMSE compared to all other models.

Next Steps

While we already tried several models, there are still some interesting future works:

1. Apply natural language processing (NLP) on Airbnb reviews for better listing price prediction.
2. Apply piecewise linear regression model.
3. Apply models on other cities or training model on other cities.

Other Potential Data Sets

- More data can be found at: Inside Airbnb.