Final Project Report

CMPT 459, Spring 2020

Written by Group Kim

Jin Young Kim	katelynk	301201074
Insoo Rhee	ira3	301075548
David Sun	david	301263907

Table of Contents

Exploratory Data Analysis

Dataset composition

The listings dataset has 15 attributes: number of bathrooms, number of bedrooms, building_id, created, description, display_address, features, latitude, listing_id, longitude, manager_id, photos, price, street_address, and interest_level. Interest_level, which is the target attribute, has three classes: low, medium, and high.

Distributions of important attributes

1) Target variable - Interest level

A histogram was created to show the distribution of interest levels (Figure 1). Most listings (~34000 listings) have low interest levels. Fewer than 5000 listings have high interest levels.

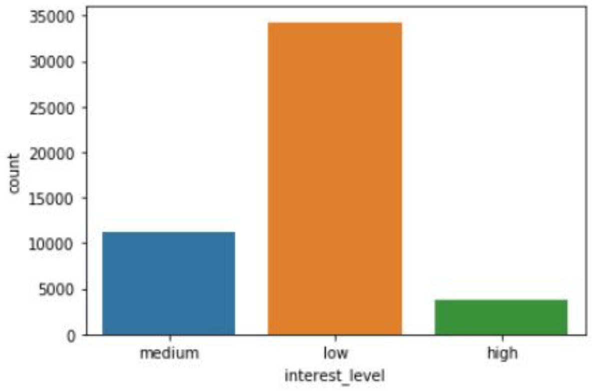


Figure 1. Interest_level distribution by count

2) Predictor variables

Histograms were created to show the distribution of each predictor variable prior to preprocessing. Only numerical variables were plotted. Below are histograms for the number of bedrooms and longitude. Whereas the histogram for number of bedrooms shows a somewhat normal distribution, that for longitude shows a single peak around the -80~-70 range. Outlier removal is needed to obtain more informative histograms. Refer to Appendix for histograms for other features.

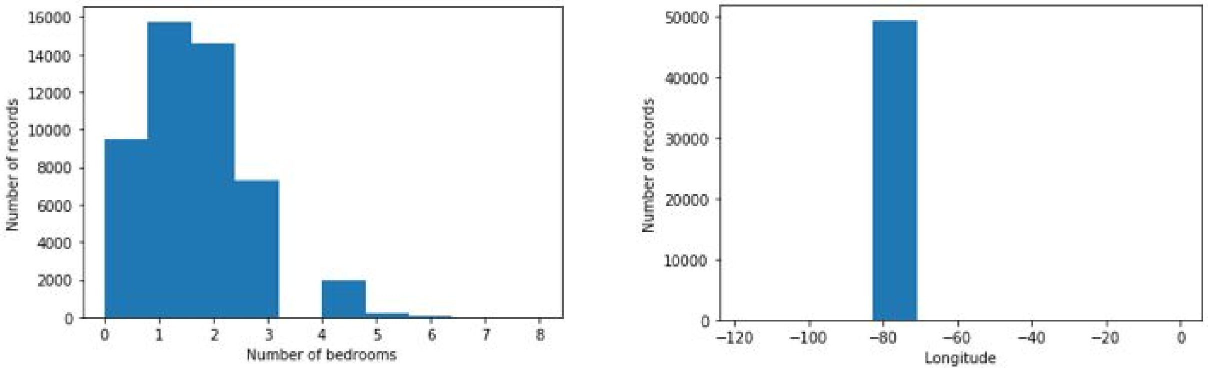


Figure 2. Histograms for number of bedrooms and longitude.

Listing trend over time

An hourly listing trend was examined by plotting the number of listings posted over time (Figure 3). The top five busiest hours of posting are from 1 a.m. to 5 a.m.

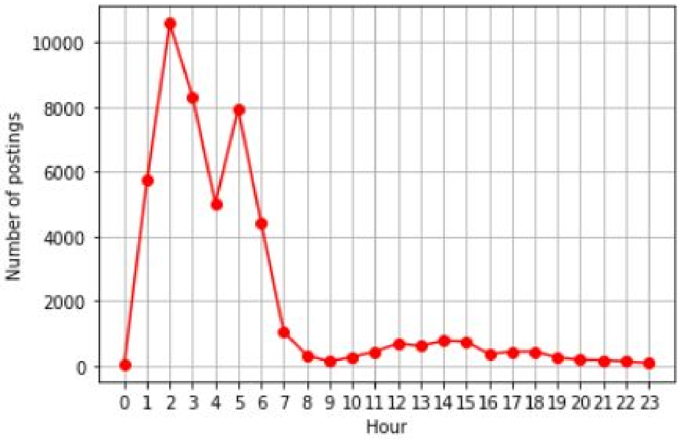


Figure 3. Rental property listing trends by hours

Data pre-processing

A. Missing values

Since missing values varied in forms between the attributes (e.g., missing values are represented as an empty list for features), we first identified these forms for each attribute. This identification was done by listing the records by their unique values for the given attribute, and then by either manually looking through the list or further sorting it in ascending order to detect any missing value. To ensure no missing values were overlooked, a further step was performed in which all possible forms of a missing value (None, " ", [ ], and 0) were searched in each column. Refer to Appendix for outlier detection results.

Missing values were detected for building_id, description, features, and photos. The missing values for these attributes cannot be imputed as they are nominal (building_id) or qualitative (rest). Given the large dataset size of ~50,000 records, it is safe to drop the missing values as we will still have a large enough sample of data to produce statistically meaningful results.

B. Outliers

Outliers were determined using the Interquartile Range Rule. The rule determines the upper and lower limits for outliers by adding IQR*1.5 to the third quartile(Q3) and subtracting IQR*1.5 from the first quartile (Q1), where IQR = interquartile range = Q3-Q1. The upper and lower limits are represented by the ends of the whiskers coming out of the box. Any data points that lie outside the whiskers are considered outliers. Boxplots of the predictor variables can be found in Appendix.

We did not perform outlier detection on features, street_address, and description because the attributes are nominal or qualitative data that are highly variable in terms of their values. Attributes such as features and description are highly variable qualitative data with little to no common values between records. For instance, it is rare to find multiple listings having the exact same description.

Histograms were created to show the distribution of the predictor variables after outlier removal. Histograms are now more informative and show a normal distribution compared to the histograms before outlier removal. Refer to Appendix for histograms of other features.

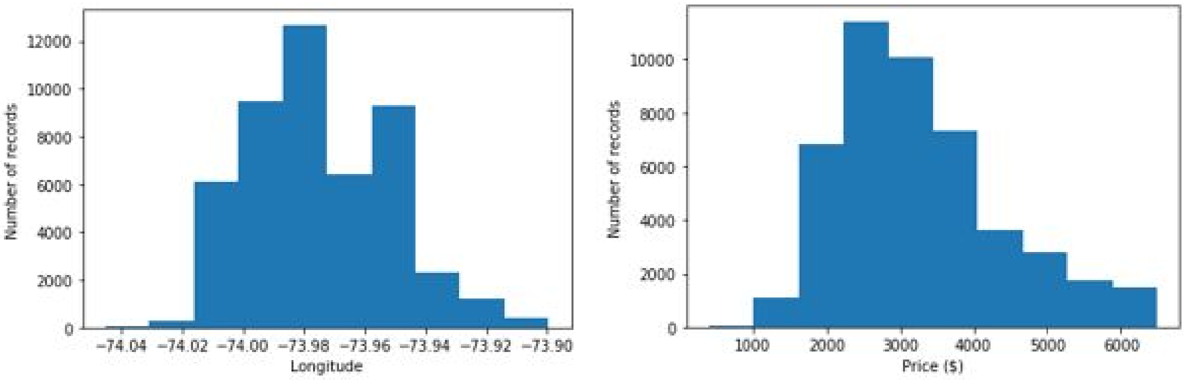
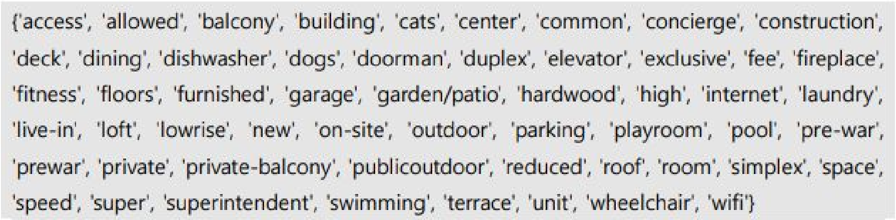


Figure 4. Histograms of longitude, latitude, and price after outlier removal

C. Feature extraction from text

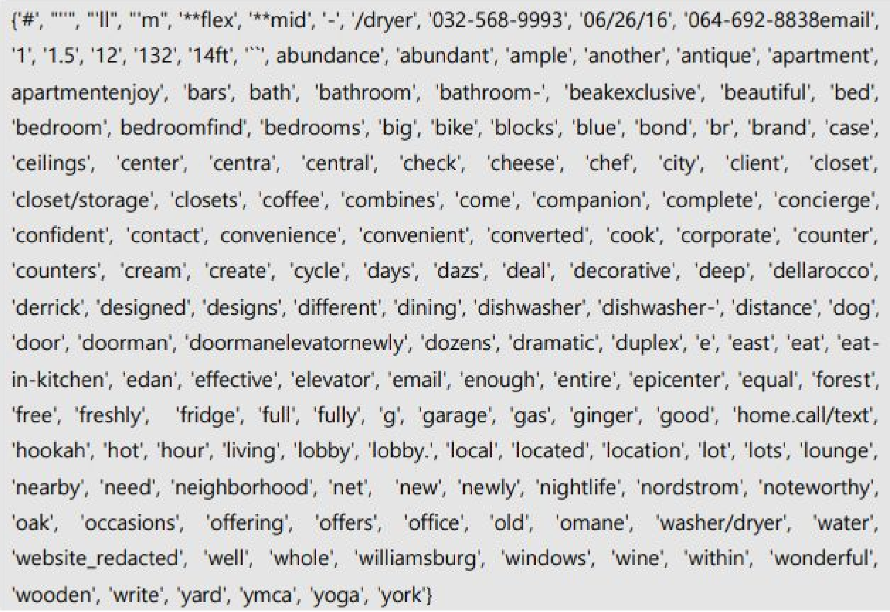
We formed a matrix containing the counts of meaningful words in our text data from the “features” and “descriptions” columns using the nltk library. We pre-processed the text data by filtering out commonly used words (e.g., “the”, “a”, “an”, and “in”). This resulted in a set of meaningful words from the records. The following is an example set resulting from preprocessing 100 data points:



```
{'access', 'allowed', 'balcony', 'building', 'cats', 'center', 'common', 'concierge', 'construction',
'deck', 'dining', 'dishwasher', 'dogs', 'doorman', 'duplex', 'elevator', 'exclusive', 'fee', 'fireplace',
'fitness', 'floors', 'furnished', 'garage', 'garden/patio', 'hardwood', 'high', 'internet', 'laundry',
'live-in', 'loft', 'lowrise', 'new', 'on-site', 'outdoor', 'parking', 'playroom', 'pool', 'pre-war',
'prewar', 'private', 'private-balcony', 'publicoutdoor', 'reduced', 'roof', 'room', 'simplex', 'space',
'speed', 'super', 'superintendent', 'swimming', 'terrace', 'unit', 'wheelchair', 'wifi'}
```

Figure 5. Text Data from Feature Attributes

Descriptions are freely written texts using a wider range of words unlike the “features” attribute, which consist of a list of words. The following is an example of a set of words from ‘descriptions’ from 10 data points.



```
{'#', "'", "'ll", "'m", '**flex', '**mid', '-', '/dryer', '032-568-9993', '06/26/16', '064-692-8838email',
'1', '1.5', '12', '132', '14ft', '``', abundance', 'abundant', 'ample', 'another', 'antique', 'apartment',
apartmentenjoy', 'bars', bath', 'bathroom', 'bathroom-', 'beakexclusive', 'beautiful', 'bed',
'bedroom', bedroomfind', 'bedrooms', 'big', 'bike', 'blocks', 'blue', 'bond', 'br', 'brand', 'case',
'ceilings', 'center', 'centra', 'central', 'check', 'cheese', 'chef', 'city', 'client', 'closet',
'closet/storage', 'closets', 'coffee', 'combines', 'come', 'companion', 'complete', 'concierge',
'confident', 'contact', convenience', 'convenient', 'converted', 'cook', 'corporate', 'counter',
'counters', 'cream', 'create', 'cycle', 'days', 'dazs', 'deal', 'decorative', 'deep', 'dellarocco',
'derrick', 'designed', 'designs', 'different', 'dining', 'dishwasher', 'dishwasher-', 'distance', 'dog',
'door', 'doorman', 'doormanelevatornewly', 'dozens', 'dramatic', 'duplex', 'e', 'east', 'eat', 'eat-
in-kitchen', 'edan', 'effective', 'elevator', 'email', 'enough', 'entire', 'epicenter', 'equal', 'forest',
'free', 'freshly',  'fridge', 'full', 'fully', 'g', 'garage', 'gas', 'ginger', 'good', 'home.call/text',
'hookah', 'hot', 'hour', 'living', 'lobby', 'lobby.', 'local', 'located', 'location', 'lot', 'lots', 'lounge',
'nearby', 'need', 'neighborhood', 'net',  'new', 'newly', 'nightlife', 'nordstrom', 'noteworthy',
'oak', 'occasions', 'offering', 'offers', 'office', 'old', 'omane', 'washer/dryer', 'water',
'website_redacted', 'well', 'whole', 'williamsburg', 'windows', 'wine', 'within', 'wonderful',
'wooden', 'write', 'yard', 'ymca', 'yoga', 'york'}
```

Figure 6. Text Data from Description Attributes

D. Feature extraction from images

The image data are jpg images of listed rooms. We extracted histogram distributions of the RGB channel, brightness (Y value), and visual words/features from the images. Using RGB channel information, we can identify patterns in the images that allow us to predict listing prices. For example, a room with wooden floor (wood color) might have a higher price than one with concrete floor (grey color). We convert the RGB color space into the YUV brightness space to obtain the Y-value (brightness) of each pixel. Brightness carries information regarding room location and orientation (a brighter room might suggest better sunlight/better lighting) and may suggest a better price.

We used OpenCV's Bag of Visual Words library to extract features from images: it processes image data and returns an array of numbers for each detected feature (green circles in the image below). In the next milestone, clustering analysis will be performed on all green circles to generate "Visual Words". Then the histogram of these "Visual Words" can help predict listing prices. For instance, stairs may suggest that a room is double-decked and hence more expensive than single-decked rooms.

E. Feature selection

A correlation heat map was created to identify attributes most relevant with interest_level. Darker grid colors (greater correlation coefficients) indicate a greater correlation. Positive correlation coefficients indicate direct relationship between the two variables, whereas negative correlation coefficients indicate inverse relationship. For instance, the correlation coefficient of 0.09 between the number of bedrooms and interest level indicates that interest level increases as the number of bedrooms increases. Listing_id, created, manager_id, and building_id attributes were not included in the correlation analysis due to their nominal nature. Additionally, street_address was not included as it provides redundant information as longitude and latitude. Based on the correlation coefficients, price most strongly correlated with interest level (-0.22) followed by bedrooms (0.09), longitude (0.041), and latitude (-0.025).

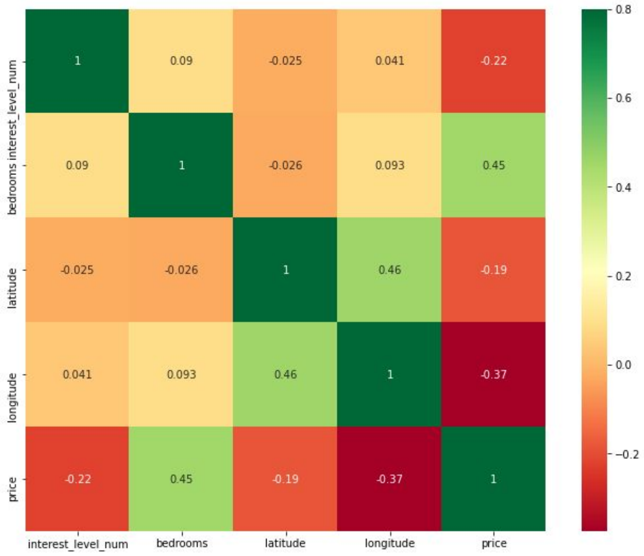


Figure 7. Correlation heat map.

Classification

A. Choice of classifiers, including choice of libraries

We used k-nearest neighbor, logistic regression, and random forest to predict listing popularity. Pandas was used for data manipulation. Scikit-learn was used to import the classifiers and measure their performance.

B. Optimization of classifiers

Three modifications were made on the classifiers: 1) adding derived features; 2) normalization; and 3) fine tuning.

1) Adding derived features

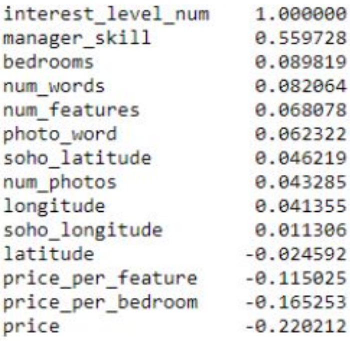
Additional features were derived from existing attributes. Table 1 lists the derived features.

Additional Attributes	Meaning
soho_longitude	Distance from Soho Mall in longitude
soho_latitude	Distance from Soho Mall in latitude
num_photos	Number of photos uploaded for the listing
num_features	Number of features listed for the listing

num_words	Number of words in the description of the listing
price_per_bedroom	Price over number of bedrooms
price_per_feature	Price over number of features
photo_word	Number of photos uploaded for the listing multiplied by number of words in the description of the listing
manager_skill	Manager competency. Calculated using the following equation: Manager_skill = high_interest_percent + 0.5 * medium_interest_percent - low_interest_ percent. The interest percentages indicate the proportion of listings of the corresponding interest level managed by a particular manager. A higher weight is given to listings with high interest levels.

Table1. Additional attributes and their meanings

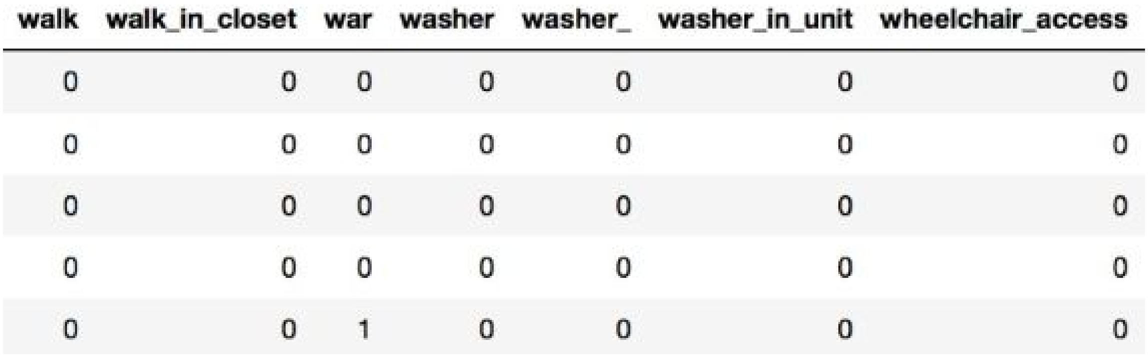
A correlation heat map was created again with the derived features added. Manager_skill had the strongest correlation with interest_level with a correlation coefficient of 0.56, followed by price (0.22), price_per_bedroom(0.17), price_per_feature (0.12), bedrooms (0.09), num_words (0.082), num_features (0.068), and photo_word (0.062). Attributes including soho_longitude(0.046), num_photos (0.043), longitude (0.041), latitude (0.025), soho_latitude (0.011) weakly correlated with interest levels and were removed from the dataset (Figure 8).



```
interest_level_num    1.000000
manager_skill         0.559728
bedrooms              0.089819
num_words             0.082064
num_features          0.068078
photo_word            0.062322
soho_latitude         0.046219
num_photos            0.043285
longitude             0.041355
soho_longitude        0.011306
latitude             -0.024592
price_per_feature    -0.115025
price_per_bedroom    -0.165253
price                -0.220212
```

Figure 8. Correlation rank with added attributes.

In addition to the derived attributes, words extracted from the “features” attribute from the original dataset were added on a column-per-word basis (Figure 9).



walk	walk_in_closet	war	washer	washer_	washer_in_unit	wheelchair_access
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	0	0	0	0

Figure 9. Words extracted from “features” are added into the dataset as new attributes.

2) Normalization

Data was normalized to ensure equal contribution by each feature.

3) Fine tuning

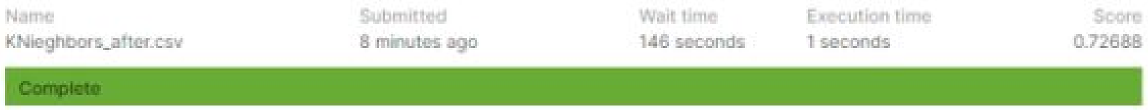
The classifiers were fine tuned by trying different values of parameters until the best performance was achieved. To help ease the tuning process, a grid search was used which selects from an input of different parameter values ones that produce the best performance. For logistic regression, the grid search was used to determine the best C parameter and regularization method (L1 and L2). Regularization filters out noise during training by penalizing large-weight coefficients. For k-nearest neighbor, the grid search was used to determine the best value for n_Neighbors (# of neighbors considered), leaf-size, and algorithm. For random forest, “max_n_estimators”, “max_depth” and “algorithm” parameters were manually adjusted until the best outcome was achieved.

C. Classifier performance

Additional optimization was performed for each classifier. Classifier performance was measured in terms of log loss on the validation set and on test dataset Kaggle.

1) K-Nearest Neighbor

The initial model without any modifications achieved log loss of 0.731 on the validation dataset and 0.72688 on the test dataset.

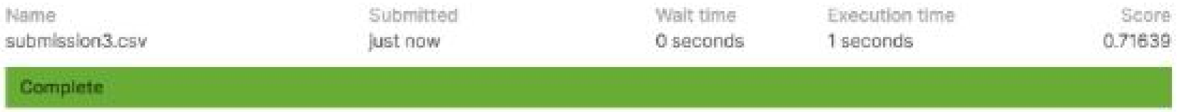


Name	Submitted	Wait time	Execution time	Score
KNieghbors_after.csv	8 minutes ago	146 seconds	1 seconds	0.72688

Complete

Figure 10. Log loss of the initial k-nearest neighbor model

The modified classifier achieved log loss of 0.721 on the validation dataset and 0.716 on the test dataset.



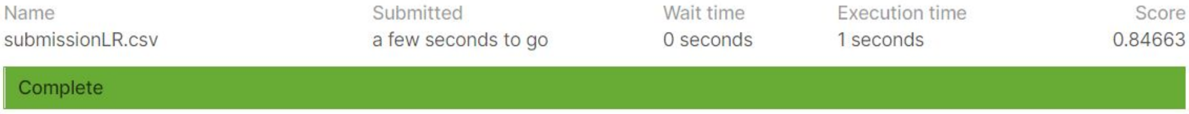
Name	Submitted	Wait time	Execution time	Score
submission3.csv	just now	0 seconds	1 seconds	0.71639

Complete

Figure 11. Log loss of the modified k-nearest neighbor model

2) Logistic regression

The initial logistic regression model (without derived attributes) achieved log loss of 0.8400 on the validation dataset and log loss of 0.8466 on the test dataset.

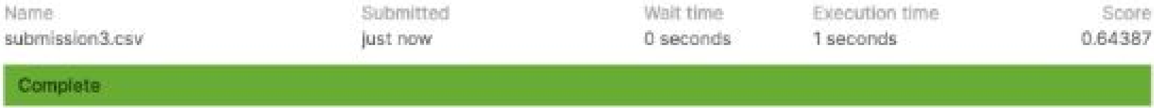


Name	Submitted	Wait time	Execution time	Score
submissionLR.csv	a few seconds to go	0 seconds	1 seconds	0.84663

Complete

Figure 12. Log loss of the initial logistic regression model on the test dataset.

The modified model achieved log loss of 0.6646 on the validation dataset and log loss of 0.6439 on the test dataset.



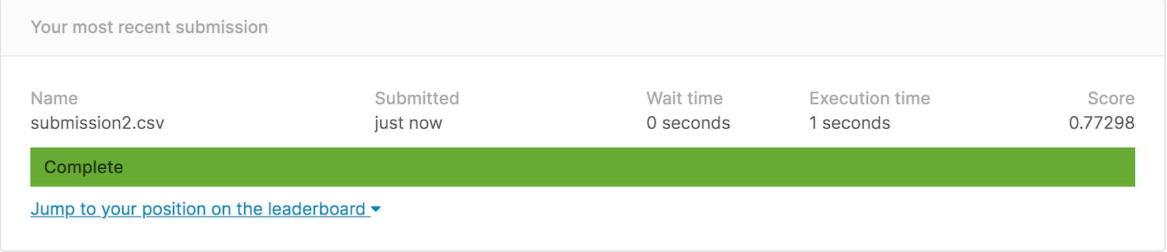
Name	Submitted	Wait time	Execution time	Score
submission3.csv	just now	0 seconds	1 seconds	0.64387

Complete

Figure 13. Log loss of the final logistic regression model on the test dataset.

3) Random Forest

The initial random forest model achieved log loss of 0.759 on the validation dataset and 0.7730 on the test dataset.



Your most recent submission

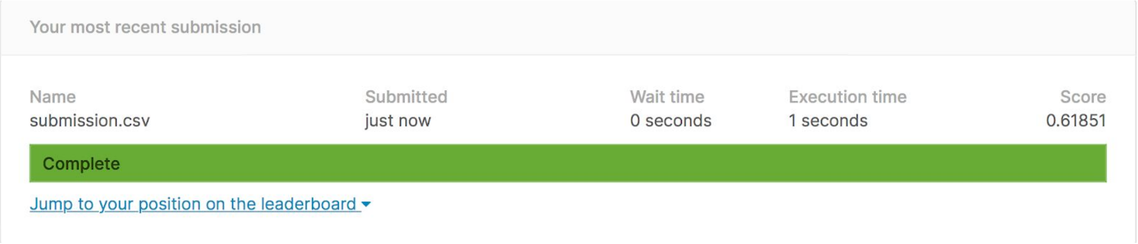
Name	Submitted	Wait time	Execution time	Score
submission2.csv	just now	0 seconds	1 seconds	0.77298

Complete

Jump to your position on the leaderboard ▾

Figure 14. Log loss of the initial random forest model on the test dataset.

The modified model achieved log loss of 0.5460 on the validation dataset and 0.6185 on the test dataset.



Your most recent submission

Name	Submitted	Wait time	Execution time	Score
submission.csv	just now	0 seconds	1 seconds	0.61851

Complete

Jump to your position on the leaderboard ▾

Figure 15. Log loss of the random forest model after modification on the test dataset.

Comparison of classifier performance

Table 2 summarizes the performance of each classifier measured in log loss.

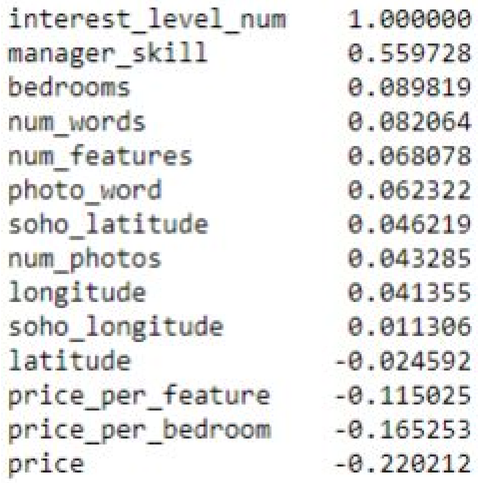
Classifier	Validation dataset	Test dataset
K-nearest neighbor	0.721	0.718
Logistic regression	0.665	0.643
Random forest	0.546	0.618

Table 2. Classifier performance (log loss) on the validation and test datasets.

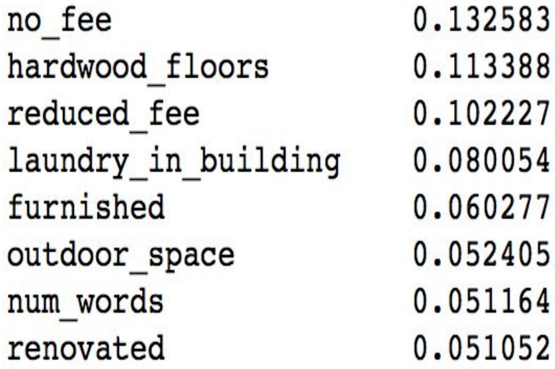
Random forest performed the best (0.618), followed by logistic regression (0.643), and K-nearest neighbor (0.718) on both the validation and test datasets.

Lessons learnt

Which features were most relevant, and why?



```
interest_level_num    1.000000
manager_skill         0.559728
bedrooms              0.089819
num_words             0.082064
num_features          0.068078
photo_word            0.062322
soho_latitude         0.046219
num_photos            0.043285
longitude             0.041355
soho_longitude        0.011306
latitude             -0.024592
price_per_feature    -0.115025
price_per_bedroom    -0.165253
price                -0.220212
```



```
no_fee                 0.132583
hardwood_floors        0.113388
reduced_fee            0.102227
laundry_in_building    0.080054
furnished              0.060277
outdoor_space          0.052405
num_words              0.051164
renovated              0.051052
```

Figure 16. Correlation rank for listing attributes

Figure 17. Top 8 features that positively correlate with interest levels.

As the correlation rank shows (Figure 16), manager_skill most strongly with interest level (0.559728), followed by price (-0.2202), price_per_bedroom (-0.1652), and price_per_feature (-0.1150). Thus, manager skills and price were the two most important attributes that contributed to a listing's popularity. To further confirm this finding, the average manager_skill was calculated for listings of each interest level. Listings of high interest levels had the highest average manager skill of 0.0352 while those of lower interest levels had negative manager skill values. Similarly, the average price was calculated for each interest level. Listings of high interest levels had the lowest average rent price ($2,589) than those of lower levels ($2,869 and $3,184). (For average calculation, refer to Appendix)

Bedrooms (0.08982), num_words (0.08206), and num_features (0.06808) positively correlated with interest levels, although the correlations were relatively weak compared to manager skill and price. These correlations show that listings with more bedrooms, and more words in the description and feature list are more likely to have high interest levels. However, one should note that a higher word count does not indicate better information quality. For instance, a listing with a lengthy but uninformative description should achieve a high interest level according to our correlation results. Furthermore, num_features can also be interpreted in two ways: a listing has a detailed feature list or a listing has many features.

Certain listing features strongly positively correlated with interest levels such as “no_fee” (0.1326), “hardwood_floors” (0.1134), and “reduced_fee” (0.1022) (Figure 17).

Which classifiers worked best, and why?

Random forest performed the best, producing the lowest log loss. Its superiority over other classifiers in our analysis may be attributed to its robustness to outliers, unbalanced data, and non-linearly separable data.

Random forest is an ensemble of decision trees. Decisions trees are robust to outliers as they prune outliers after splitting on non-outlier data points first. On the contrary, logistic regression and k-nearest neighbors are sensitive to outliers. In logistic regression, outliers have the same impact on the model as all the other data points. In k-nearest neighbors, a single outlier can significantly change the class boundary of an object especially for smaller k’s. Decision trees are also non-linear in nature unlike logistic regression, which assumes that data is linearly separable. Given the complexity and large feature space of our dataset, it is unlikely that our data is linearly separable. Lastly, decision trees are robust to skewed data given a large enough tree depth [1]. With careful hyperparameter tuning of the tree depth, we can achieve classifier performance at which outliers and skewed data are well-handled. On the contrary, K-nearest neighbors are sensitive to skewed data. If class A is much more common than class B, KNN gives a higher preference to A than B, which can lead to misclassification of a class B object as class A. As seen in the exploratory analysis, the training dataset was highly skewed with around 70% of the listings labeled as “low” interest level. The same trend may exist in the test dataset and may have contributed to the relatively poor performance of the k-nearest neighbors.

Which classifiers were more efficient to train?

Table 3 shows the time complexity of each classifier. Random forest has the highest time complexity, followed by logistic regression and k-nearest neighbor. K-nearest neighbor was the most efficient to train.

Classifier	Time complexity
K-nearest neighbor classifier	O(N * K)
Logistic regression	O(N* K * 100 (default epoch/num of ierations) * 3 (# of classes)) = O (300 * N * K)
Random forest	O (N * K^2 * M)

M = number of estimators (~700), N = number of training examples, K = number of features

Table 3. Time complexity of classifier [2, 3, 4]

Was overfitting a problem? If so, how did you address it?

Overfitting was a problem during classification. Overfitting was checked by comparing between the training accuracy and test accuracy measured in terms of log loss. If the model achieved a higher accuracy on the training dataset than the test dataset, the model was deemed overfitting.

Overfitting was addressed in the preprocessing and classification phases. In the preprocessing phase, we limited the attributes to more informative, numerical attributes including price, longitude, latitude, number of bedrooms, number of bathrooms. Nominal attributes such as listing_id and building_id were removed. Next, we removed outliers based on the interquartile range and removed rows with missing data. In the classification phase, all classifiers were optimized by adding derived attributes such as manager_skill and price_per_bedroom. Feature attributes from text extraction were also added. Attributes that weakly correlated with interest level such as soho_latitude and longitude were removed from the dataset. Lastly, hyperparameter tuning was performed to improve the classifier.

While the above methods improved the classifiers' performance, overfitting still persisted in the modified random forest. Table 4 shows which models overfit.

Classifier	Validation dataset	Test dataset	Overfitting
Initial k-nearest neighbor	0.731	0.72688	No
Modified k-nearest neighbor	0.721	0.716	No
Initial logistic regression	0.8400	0.8466	Yes
Modified logistic regression	0.6646	0.6439	No
Initial random forest	0.759	0.7730	Yes
Modified random forest	0.5460	0.6185	Yes

Table 4. Overfitting

Recommendations for rental property owners

Based on our correlation analysis results, we recommend property owners to highlight the following attributes when promoting existing properties or ensure their new properties have the following attributes in the order of highest importance to lowest:

1. Manager competency - Manager_skill most strongly and positively correlated with interest levels; a competent manager draws more renters.
2. Good (= cheaper) price - Price, price_per_bedroom, and price_per_feature were the next most strongly correlating features after manager_skill. They negatively correlated with interest levels; cheaper prices attract more renters.
3. Important features - Certain features such as no_fee and laundry_in_builiding positively correlated with interest levels. Owners must highlight these features if their existing properties have them or ensure their new properties will have these features.
4. Detailed description and feature list - num_word and num_features weakly positively correlated with interest levels. The longer the description and feature list, the higher the interest level. Property owners can consider writing a detailed description and feature list to make sure potential renters gain full information about the properties' benefits.

References

Data science work-flow and feature engineering

gdy5. (2017, June 5). *CV statistics (Better parameters and explanation)* .

Retrieved from Kaggle:

https://www.kaggle.com/guoday/cv-statistics-better-parameters-and-explaination

Manah, S. (2019, April 14). *Titanic Data Science Solutions*. Retrieved from Kaggle: https://www.kaggle.com/startupsci/titanic-data-science-solutions

Model comparison

[1] Muchlinsk, "Comparing Random Forest with Logistic Regression for Predicting Class-Imbalanced Civil War Onset Data", *Political Analysis*, October 22, 2017, Available on: http://davidsiroky.faculty.asu.edu/PA2016.pdf

[2] Maszke, "*What is the Search/Prediction Time Complexity of Logistic Regression?"*, January 17, 2019. Retrieved from Stack Overflow: https://stackoverflow.com/questions/54238493/what-is-the-search-prediction-time-complexity-of-logistic-regression

[3] Su, "A Fast Decision Tree Learning Algorithm", *American Association for Artificial Intelligence*, J., 2006, p.60-61.

[4] Veksler, "*CS840a Machine Learning in Computer Vision".* Retrieved from Stack Exchange: http://www.cs.haifa.ac.il/~rita/ml_course/lectures/KNN.pdf

Distribution of Tasks Within Team

Jin Young Kim

- Classification, lessons learned, recommendations, whole paper revision.

Insoo Rhee

- Exploratory Data Analysis
- Recommendations for rental property owners

David Sun

- Random forest ,feature engineering and research for better performance