MIDTERM

Nikhil Agarwal
PREDICT 413 | Section 57
Kaggle screenname: NikAgarwal

Table of Contents

PROBLEM DESCRIPTION 4

SIGNIFICANCE 4

OVERVIEW OF DATA 4

DATASETS 4
NEW VARIABLE CREATION 5
DESCRIPTIVE STATISTICS 5
VISUAL DATA EXPLORATION 6
OUTLIER MANAGEMENT AND DATA CLEANSING 8

LITERATURE 9

KNN 9
RANDOM FORESTS 9
XGBOOST 9
LINEAR DISCRIMINANT ANALYSIS (LDA) 10
QUADRATIC DISCRIMINANT ANALYSIS (QDA) 10

TYPE OF MODELS 10

EXTREME GRADIENT BOOSTING (XGBOOST) 11
RANDOM FORESTS 11
K-NEAREST NEIGHBOR (KNN) 11
LINEAR (LDA) AND QUADRATICS DISCRIMINANT ANALYSIS (QDA) 11

FORMULATION 12

PERFORMANCE/ACCURACY 12

LIMITATIONS 13

FUTURE WORK 13

TEXT ANALYTICS 14
OUTLIER MANAGEMENT 14
GPS COORDINATE EXPLORATION 14
EXTERNAL DATA SOURCES 14

LEARNING 14

WORKS CITED 16

APPENDIX 17

DATA DICTIONARY 17

VARIABLE MGR_SKILL DEVELOPMENT AND EXPLANATION 17
FREQUENCY OF LISTINGS BY DAY OF WEEK 18
FREQUENCY OF LISTINGS BY HOUR OF DAY 18
FREQUENCY OF LISTINGS BY DAY (DATE) 19
CONFUSION MATRIX & ACCURACY OUTPUT FOR XGBOOST MODEL 20
CONFUSION MATRIX & ACCURACY OUTPUT FOR RANDOM FOREST MODEL 21
CONFUSION MATRIX & ACCURACY OUTPUT FOR KNN MODEL 22
CONFUSION MATRIX & ACCURACY OUTPUT FOR LDA MODEL 22
CONFUSION MATRIX & ACCURACY OUTPUT FOR QDA MODEL 23

Problem Description

The intent of this activity was to predict the interest level of potential customers researching places to live. Renthop is a website that hosts rental listings in many major cities across the United States. For this activity, many of the rental listings were from the New York City boroughs. This dataset was provided by Kaggle.com and featured 14 different potential predictor variables. Several statistical models were constructed for this activity. This report will discuss five of the models constructed using K-Nearest Neighbors (KNN), Extreme Gradient Boosting (XGBoost), Random Forests, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA). Since the results were posted in Kaggle, a summary of the log-loss scores for each submitted model through the Kaggle competition is provided in the Performance/Accuracy section. Prior to submission, confusion matrices and the accuracy for each model was assessed and is discussed in the Performance/Accuracy section.

Significance

Each year, millions of potential tenants' search for rental properties to move into. Renthop is an innovative service that enables renters to find properties that match their interests and requirements quickly. For the rental companies (and landlords), Renthop is an excellent way to showcase their properties and minimize the amount of time a property remains unoccupied. The significance of this activity is to identify opportunities that will enable Renthop to highlight sub-par information that may result in less interest for a property. According to Kaggle.com, this activity "will help Renthop better handle fraud control, identify potential listing quality issues, and allow owners and agents to better understand renters' needs and preferences" (Kaggle, 2017).

Overview of Data

Datasets

Two key datasets were provided by Kaggle. The first was the training dataset and the second was the test dataset. The key difference between the training and test datasets was the presence of the response variable, *interest_level*, in the training dataset but not the test dataset. Both datasets were provided in JSON format and upon import to R, was converted to a dataframe. Table 5 (in APPENDIX) provides a summary of the imported predictor variables and their descriptions. The training dataset was further split, using a 70%-30% ratio, to create a validation dataset. This ensured that models developed using the new dataset could be validated with the validation dataset prior to operationalizing with the test dataset. Recall that the test dataset does not have any response variable and the goal is to develop the probability of the potential response (for each possible response of high, medium, or low).

New Variable Creation

Text analytics was not used in any of the exploratory or modeling steps. Rather, new variables were constructed to help summarize some of the attributes of the text-only columns (i.e., description, photos, and features). For instance, a new variable was constructed that counted the number of words found in the description. Another variable was constructed that counted the number of characters in the description. A variable was also developed that looked at the total number of features listed for each listing.

Table 1 outlines the derived variables.

Table 1: List of Derived Variables

Derived Variable Name	Description
total_pix	Counts the number of photos for each listing
total_features	Counts the number of features listed for each listing
total_description_length	Counts the number of characters in the description column
total_words	Counts the total number of words in the description column
price_per_room	Rental price divided by the total number of bedrooms and bathrooms
price_per_bedroom	Rental price divided by the total number of bedrooms
real_address_chk	Compares the street address and display address. If they match, this value is 1 otherwise 0
mgr_skill	Metric that calculates a score for manager IDs based on the interest of each of their respective listings
create_hour	The hour listing was created (on 24-hour)
create_day	The day of the week (e.g., Friday, Saturday, etc.)
create_daynumber	The numeric day

A key change for the model was to have the interest_level values to be considered as factors. This ensured that all models constructed yielded an answer that was one of the three possible responses[1] (for the response variable). An interesting metric, mgr_skill, was developed to help gauge a manager's ability to 'showcase' the property. See APPENDIX for explanation of the development of this metric.

It was discovered that the provided training and test datasets had overlapping time periods. For instance, both datasets covered the months of April through June. Therefore, it was decided to parse the date and time stamp of the *created* variable. With this parsing, three new variables were constructed (as seen in Table 1): *create_hour*, *create_day*, and *create_daynumber*.

Descriptive Statistics

Table 2 provides a brief summary of the numeric variables in the training (after the 70/30 split) dataset. Note the strong presence of outliers in almost each variable.

[1] The response variable, *interest_level*, has three possible responses: low, medium, high.

Table 2: Descriptive Statistics on Numeric Variables

Variable	Min	Median	Mean	Max
bathrooms	0	1	1.215	10
bedrooms	0	1	1.549	8
price	43	3150	3902	4490000
total_pix	0	5	5.62	68
total_features	0	5	5.429	39
total_description_length	0	564	602.5	3367
total_words	0	86	93	567
price_per_room	43	1300	1600	1496667
price_per_bedroom	43	2150	2491	2245000
mgr_skill	0.01327	0.34483	0.40542	1.88235

Table 3 provides a brief overview of the count of interest levels within the training dataset.

Table 3: Frequency of interest level for training dataset

Interest Level	Frequency
high	2665
medium	7881
low	24000

Table 6, Table 7, and Table 8 (in the APPENDIX) provide a frequency summary of the respective variables. Note that each of these variables were classified as factors. The reasoning was that perhaps there's a significance to the day of when listings are created or perhaps an interest level. For instance, some rental properties (in general) allow for rent payments to be made twice a month, instead of the more common, first of the month. Furthermore, the hour of the day was interesting to include since the analysis could then explore the significance of the hour of listing creation.

Visual Data Exploration

Numerous visual charts were created to help explore the data visually. However, for succinctness, only a few have been included and discussed in this paper. Figure 1 breaks down the variable *mgr_skill* by interest level. It's clear to see that low interest level also tends to have a smaller IQR (inter-quartile range) than high or medium interest levels. Note how the median for the low interest level is below that of the medium and high interest levels.

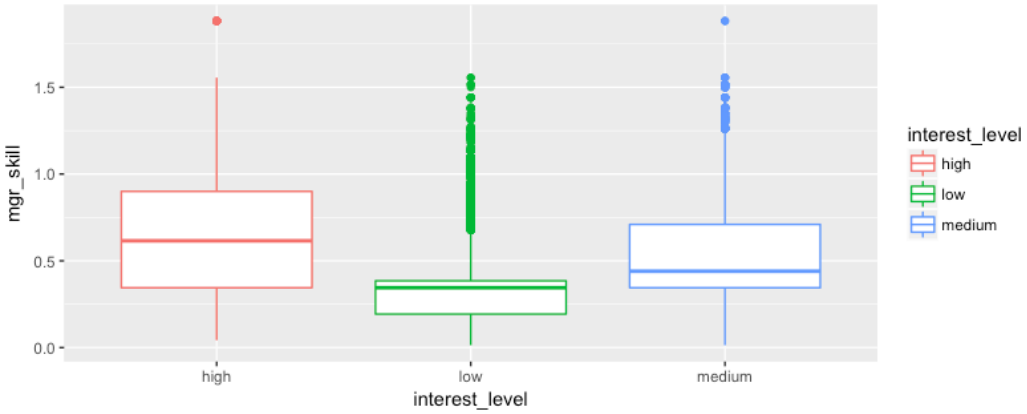

Figure 1: mgr_skill vs. interest_level boxplot

Figure 2 breaks down the median rental price by interest level and day of the week. This chart helps clarify the median listing price by day of week for each interest level and how it varies. For instance, the median price for any day of the week with high interest tends to be $2,500 or less. In contrast, with low interest levels, the median price is well above $3,100. Interestingly, with medium interest, the median price tends to be between $2,750 and $3,000. Note how the median price decreases (for medium interest level) after Tuesday, plateauing on Friday, and then sharply increasing on Saturday. It is possible that there's stronger interest in properties on Saturday (since more people have more free time to browse Renthop). The same pattern is not necessarily there for the other interest levels.

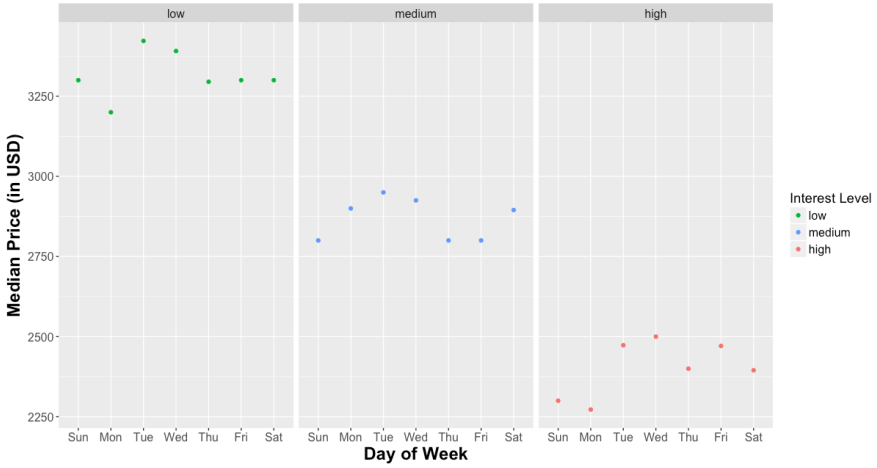

Figure 2: Median rental price by day of week and interest level

Figure 3 is a slight twist on what was seen in Figure 2. In this case, a grouping called Time of Day was created based on the hour of listing creation time. Different hours of the day were classified[2] as morning, afternoon, evening, or late night. From this, a median rental price was

[2] Morning hours were 06:00 to less than 12:00, Afternoon hours were 12:00 to less than 18:00, Evening hours were 18:00 to less than 23:00, Late night hours were 23:00 to less than 06:00.

calculated and separated by interest level. Note how if a property has high interest, it's median price is lowest during the afternoon hours. This contrasts with both low and medium interest levels as their median prices were lowest during evening hours.

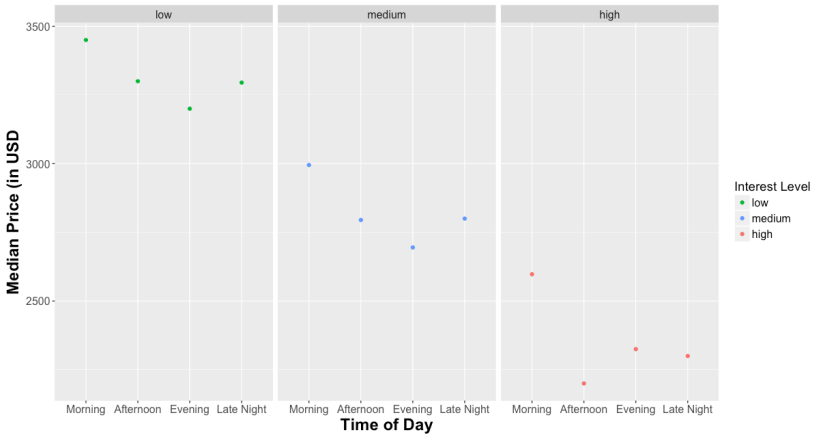

Figure 3: Median rental price by time of day and interest level

Figure 4 illustrates the average number of words in the description by time of day and separated by interest level. In all three interest levels, it appears that the most number of words are occurring in the morning and late night hours. Intuitively, it makes sense as the managers making the listings have more time to type in descriptions in the early and late hours since there may not be as many showings or visitors. It's also clear to see that the high interest properties also tend to have more words on average than the low interest properties.

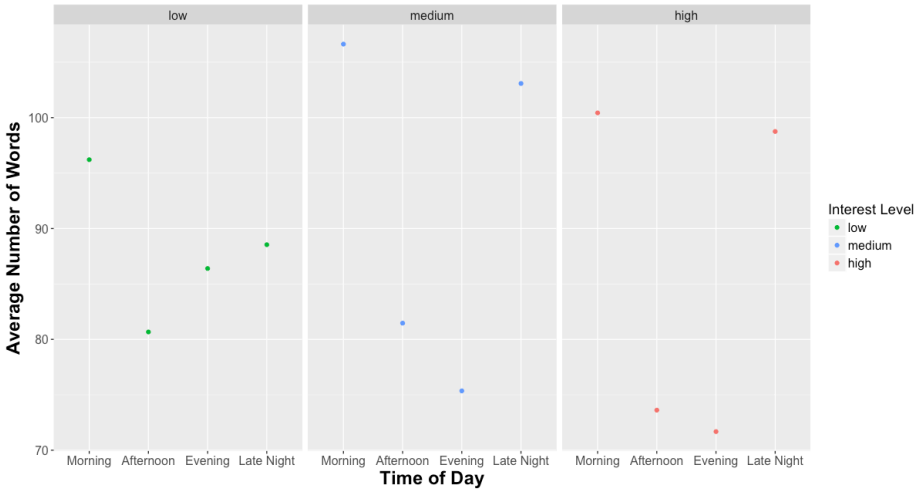

Figure 4: Average number of words by time of day and interest level

Outlier Management and Data Cleansing

For this iteration of the analysis, no outlier management techniques were undertaken. The primary reason for this was to develop models with actual values in place and understand their performance. Outlier management will be reserved for future steps. A check was made for missing data and none were found. Furthermore, observed values in the training dataset were not transformed nor were they imputed with other values to minimize the effects of erroneous or outlier values.

Literature

This analysis is focused on predicting (i.e., forecasting) the interest level for a potential rental listing. Recall from earlier discussions that the interest levels are actually factors and that this analysis becomes a simple classification example. In the previous section (Type of Models), five different models were selected. This section discusses academic (from peer reviewed journals) papers that utilize similar models for classification.

KNN

In the paper, "Large scale biomedical texts classification: a kNN and an ESA-based approaches", the authors discuss two methods of classifying texts into several categories. Ideally, they want their model to perform with unseen texts with high accuracy. Their "kNN-based approach requires a collection of documents previously annotated for the neighbours' retrieval. For a given document, the aim is to retrieve its k most similar documents" (Dramé, Mougin, & Diallo, 2016). This methodology is very similar to constructing a training dataset and determining what the current classifications are to help the model learn and then apply to unknown values or test dataset. Ultimately, in their research, the authors found KNN to be a very good model and also enhanced it with random forests.

Random Forests

In the paper, "Crop classification of upland fields using Random forest of time-series Landsat 7 ETM+ data", the authors discuss using crop classification through the use of random forests. An interesting approach that the authors used was satellite data with a time series focus. Although the analysis in this Kaggle competition did not require a time series approach, the use of external data sources (discussed in FUTURE STEPS) could be a great way to augment the random forest model's accuracy. One of the key conclusions the authors made is the random forest model's ability to help detect false classifications and still ensure fairly high accuracy (Tatsumi, Yamashiki, Torres, & Taipe, 2015).

XGBoost

In the paper, "Bioactive Molecule Prediction Using Extreme Gradient Boosting", the authors discuss "(Xgboost)…was investigated for the prediction of biological activity based on

quantitative description of the compound's molecular structure" (Mustapha & Saeed, 2016). Interestingly, the authors conduct a direct comparison to other ensemble models and neural network models and distinctively conclude the superior performance of the XGBoost model. From an accuracy standpoint, their research concluded that XGBoost had over 94% accuracy on their datasets and outperformed the Random Forest model.

Linear Discriminant Analysis (LDA)

In the paper, "Classification of rice wine according to different marked ages using a novel artificial olfactory technique based on colorimetric sensor array", the authors describe how they used an LDA model to classify RGB (red, green, blue) components. Their primary focus was comparing PCA (principal component analysis) and LDA. The PCA model that they constructed was actually decent, however, the distinctiveness of the class groupings found by the LDA model was exceptional. Figure 5 (Ouyang, Zhao, Chen, & Lin, 2012) illustrates the groupings between the two models. Note how closely clustered the LDA model found its groupings. In contrast, the PCA model is subject to misclassification.

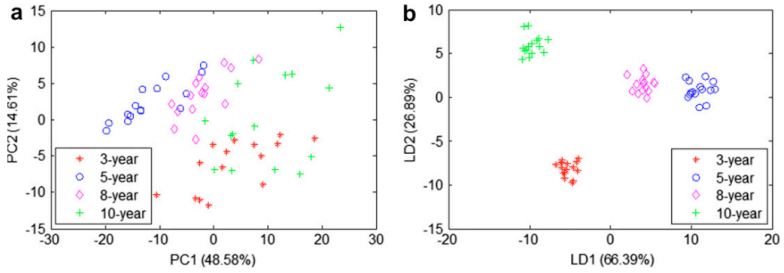

Fig. 4. Score plots of PCA analysis (a) and LDA analysis (b) for rice wine of different marked ages.

Figure 5: LDA vs. QCA (Ouyang, Zhao, Chen, & Lin, 2012)

Quadratic Discriminant Analysis (QDA)

In the paper, "Environmental Noise Classification using LDA, QDA and ANN Methods", the authors attempt to build LDA, QDA, and ANN models to classify environmental noise. The interesting approach in their work is their direct comparison of LDA and QDA (for this brief write-up, ANN models are disregarded). Their research showed that a QDA model did out perform an LDA model. Nevertheless, the underlying message in their research shows that construction of different models is necessary to truly understand the significance of a model.

Type of Models

Several different types of models were constructed, but only five of these models were chosen to be submitted on Kaggle. All five models can be considered as machine learning models. The five models are:

- Extreme Gradient Boosting (XGBoost)
- Random Forests
- K-Nearest Neighbor (KNN)
- Linear Discriminant Analysis (LDA)
- Quadratic Discriminant Analysis (QDA)

Extreme Gradient Boosting (XGBoost)

The XGBoost model was chosen due to its veracity and its complex learning abilities. Recall from the VISUAL EXPLORATION section that there appears to be numerous behaviors that affect interest level. Decision trees and boosted models are able to capture nuances in the datasets and produce accurate models.

Random Forests

Random forests is an interesting approach as this machine learning staple creates 'tiny' trees that are not very deep (in terms of branches). Furthermore, the algorithms create thousands of trees (each one could virtually use any number of predictors and no tree necessarily has all possible predictors) very quickly. According to the authors of "An Introduction to Statistical Learning with Applications in R", "random forests [are] an improvement over bagged trees by way of a small tweak that decorrelates the trees" (James, Witten, Hastie, & Tibshirani, 2015). This is a clear reason why bagging models were not considered since bagging models will enable the use of all predictors in a tree.

K-Nearest Neighbor (KNN)

KNN models are somewhat simple and utilize Bayes classifiers. Recall that the variable *interest_level* is a factor with three values (high, medium, low). At first glance, the probability of a listing being one of those three is not known. Distinctively, "KNN applies Bayes rule and classifies the test observation…to the class with the largest probability" (James, Witten, Hastie, & Tibshirani, 2015). Surprisingly, the KNN model can be quite accurate yet easy to explain. Compared to the decision tree models (XGBoost and random forests), KNN models can be computationally inexpensive.

Linear (LDA) and Quadratics Discriminant Analysis (QDA)

LDA and QDA models are quite popular for multiple class classification modelling. Since this analysis is an exemplary example of multiple classification, both LDA and QDA are appropriate. The key difference between LDA and QDA is the fact that the QDA approach assumes that "each class has its own covariance matrix" (James, Witten, Hastie, & Tibshirani, 2015). Both modelling techniques assume that the classifications are from a Gaussian distribution. However, such information is not necessarily known with the provided datasets. Therefore, the author of this analysis is not confident in the performance of these models.

Formulation

All the models were constructed using R[3]. The following are the key packages used within R:

- Caret
- MASS
- XGBoost

The caret package is quite powerful and enables easier cross-validation for any type of model. One of the control parameters for caret is the trainControl function. With this control parameter, XGBoost, random forest, and the KNN models utilized repeated cross validation, using 10 folds and repeated three times. The cross-validation approach enables a stronger learning capability for the models since a small subset of the training data can be used for validation prior to the validation dataset being used. Since this is done inherently within the caret package, the final model can be considered a decent model with the given parameters.

For the LDA and QDA models, the library MASS was used. For these two models, no cross validation was done.

As mentioned earlier, the original dataset from the Kaggle competition was subdivided into a 70%-30% split. In other words, 70% of the data were used to 'train' the models. After that, the 30% of the data were used as a validation dataset to validate the trained models. Once the model accuracy and confusion matrices were satisfactory, the trained model would then be applied to the test dataset that was provided by Kaggle. The 30% of the data were not reintroduced to the model as that would constitute a potential leakage that could result in false perceptions.

Some of the provided information was also not utilized in the model exploration or development phases. For instance, the images were not explored for their content. However, the number of images was used.

Performance/Accuracy

In order to assess the models, a confusion matrix was created for each model to compare the model's performance on the validation dataset. All models were also assessed for their accuracy (i.e., mean square error). Furthermore, all five models discussed in this analysis were submitted to Kaggle to obtain a log-loss score. See the APPENDIX for the confusion matrices for each model.

[3] See the R code to see the exact syntax for each model.

Table 4 summarizes the results of each of the models

Table 4: Model accuracy & Kaggle score summary table

Model	Accuracy	Kaggle Private Score	Kaggle Public Score
Random Forest	0.7379	0.60147	0.60351
KNN	0.7069	0.64217	0.86642
LDA	0.7082	0.71016	0.72011
QDA	0.2616	4.22534	4.23345
XGBoost	0.7471	0.57548	0.57727

The Kaggle Public Score was found using only 10% of the dataset to assess the test model. The Kaggle Private Score was found using all of the data in the test dataset and was only available after the competition closed. In both instances, the lower the score, the better the model.

Based on the models developed and submitted to Kaggle, the XGBoost model outperformed all others. However, the next closest model was random forest. Note how the KNN model, which is simple, was third best. Surprisingly, the QDA model was terrible compared to the other four models.

Limitations

There are two key limitations to the models constructed complexity and optimization. Both random forest and XGBoost are complex models that are not very easy to explain. For instance, both approaches constructed thousands of trees, but there is no distinctive equation that can be documented and easily explained. Rather, the explanation of each tree would result in a convoluted explanation of the logic and the randomness.

Furthermore, both models were computationally expensive. For instance, the XGBoost model utilized 32 cores for a run time of over 75 minutes. One key area to explore would be the optimization of each of these algorithms. For instance, the shrinkage parameter would enable the reduction of predictors and could improve a fit of a model while also reducing the likelihood of overfitting a model. Although it is difficult to overfit a model using boosting methods, it is still possible and the choice of parameters could affect the accuracy.

Future Work

There are four key areas of improvement for the next steps:

- Text analytics
- Outlier management
- GPS coordinate exploration
- External data sources

Text Analytics

For this analysis, a deep dive into the context of the description and listed features was not done. Exploring these two areas in terms of text would enable a greater understanding of the type of features and keywords that generate stronger interest. Another analysis to incorporate would be sentimental analysis to help gauge an 'excitement' factor that may encourage users to click on a listing and potentially drive more interest.

Outlier Management

As previously mentioned, the presence of outliers was quite glaring. The risk with elimination of outliers is that the model may work well with both training and validation datasets, it may not work as well with the test dataset. Furthermore, the outliers themselves need a greater understanding. It may be possible that there are different types of rental postings such as postings catering to wealthier clientele. One of the benefits of machine learning algorithms is their inherent ability to account for outliers. However, if outliers are managed properly, these algorithms may experience improvements in their accuracy.

GPS Coordinate Exploration

Although the GPS coordinate was used as is in this analysis, it may be prudent to explore the GPS coordinates more. One glaring error in the dataset was the presence of 0 in both latitude and longitude. For reference, a latitude of 0 and a longitude of 0 suggests the location is in the middle of the ocean off the coast of Africa. Therefore, it may be prudent to either impute these values or determine the GPS coordinate for these locations based on street address.

External Data Sources

Finally, more data exploration would be prudent to develop other key metrics as necessary. The *building_id* variable, for instance, has been repeated multiple times and it may allude to interest level amongst Renthop visitors. Examples such as this are a great segue way in to conducting deeper analysis of the available information. One of the restrictions in this competition through Kaggle was the inability to bring in external data sources. External data sources such as weather or temperature may also drive other insights that could influence interest level.

Learning

Perhaps the best part about this assignment was the use of JSON files. I have never directly worked with JSON files and being able to parse them for information is astounding. Another key benefit of this assignment was the ability to put my knowledge on machine learning models

(e.g., random forest, boosting, etc.) to the test with real world data. In the machine learning class, we used academic datasets in which the answers were somewhat known. In this assignment, since we did not have the known final answers, it was quite exhilarating to develop models and compete.

Finally, a key takeaway for me is the importance of data exploration. I spent a considerable amount of time reading the different forum posts on Kaggle to better understand how some of the variables worked or what they actually meant. In hindsight, I easily could have spent another eight weeks just exploring the data and developing new variables to help improve the models.

Works Cited

Dramé, K., Mougin, F., & Diallo, G. (2016). Large scale biomedical texts classification: a kNN and an ESA-based approaches. *Journal of Biomedical Semantics*, 1-12.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2015). *An Introduction to Statistical Learning with Applications in R* (6th Edition ed.). New York: Springer Science + Business Media.

Kaggle. (2017). *Two Sigma Connect: Rental Listing Inquiries*. Retrieved April 26, 2017, from Kaggle: https://www.kaggle.com/c/two-sigma-connect-rental-listing-inquiries#description

Mustapha, I. B., & Saeed, F. (2016). Bioactive Molecule Prediction Using Extreme Gradient Boosting. *Molecules, 21*, 1-11.

Ouyang, Q., Zhao, J., Chen, Q., & Lin, H. (2012, December 5). Classification of rice wine according to different marked ages using a novel artificial olfactory technique based on colorimetric sensor array. *Elsevier Food Chemistry*, 1320-1324.

Tanweer, S., Mobin, A., & Alam, A. (2016, September). Environmental Noise Classi cation using LDA, QDA and ANN Methods. *Indian Journal of Science and Technology, 9*, 1-8.

Tatsumi, K., Yamashiki, Y., Torres, M. A., & Taipe, C. L. (2015, June 14). Crop classification of upland fields using Random forest of time-series Landsat 7 ETM+ data. *Elsevier Computers and Electronics in Agriculture*, 171-179.

APPENDIX

Data Dictionary

Table 5: Data dictionary

Column	Description
bathrooms	Number of bathrooms
bedrooms	Number of bedrooms
building_id	Unique identifier for location of property
created	Date listing created
description	Text describing the property (written by the creator of the listing on Renthop)
display_address	Address of property
features	List of features of the apartment
latitude	Latitude of the address
longitude	Longitude of the address
listing_id	Unique identifier for each listing.
manager_id	Unique identifier for each manager
photos	List of photos posted (URLs)
price	Price of rental property (in USD)
street_address	Address of property
interest_level	Target (response variable): high, medium, low

Variable mgr_skill development and explanation

The variable, mgr_skill, was developed as a metric to gauge the effectiveness of a manager tied to a listing. Note that this variable was created using the data found in the training dataset after the 70% split from the provided training data. It was NOT updated afterwards using the 30% of the data in the validation dataset. This was done to ensure that there was no 'leakage' of information in the modelling process.

For each unique manager ID, a count of all their listings along with the total number of listings in each potential interest level (i.e., high, medium, low) was constructed. Then a ratio was calculated (e.g., total number of high interest level properties divided by total number of listings). Two points (the scoring was arbitrary) were multiplied by the ratio of total high interest over total properties and one point was multiplied by the ratio of total medium interest over total properties. The equation below describes the formula:

$$mgr\ skill = 2 * \frac{total\ number\ of\ high\ interest\ level\ properties}{total\ number\ of\ properties\ listed\ for\ manager} + 1 * \frac{total\ number\ of\ medium\ interest\ level\ properties}{total\ number\ of\ properties\ listed\ for\ manager}$$

To reduce the likelihood of penalizing managers with few listings, any manger with less than 10 total properties listed will be given the median score (approximately. 0.344) of all the manager skills.

Frequency of listings by Day of Week

Table 6: Frequency of listings by day of week

Day of Week	Frequency
Sunday	3147
Monday	2997
Tuesday	5847
Wednesday	6610
Thursday	5723
Friday	5335
Saturday	4887

Frequency of listings by hour of day

Table 7: Frequency of listings by hour of day

Hour	Frequency	Hour	Frequency
1	4002	*13*	408
2	7356	*14*	525
3	5885	*15*	525
4	3597	*16*	243
5	5544	*17*	318
6	3099	*18*	316
7	735	*19*	193
8	232	*20*	135
9	94	*21*	127
10	205	*22*	98
11	325	*23*	60
12	486	*0*	38

Frequency of Listings by Day (date)

Table 8: Frequency of listings by day (date)

Day Number	Frequency	Day Number	Frequency
1	641	*16*	1278
2	1238	*17*	1101
3	1244	*18*	1037
4	1059	*19*	988
5	1274	*20*	1236
6	1298	*21*	1728
7	1255	*22*	1180
8	1191	*23*	758
9	1054	*24*	1274
10	1112	*25*	1047
11	1224	*26*	1030
12	1631	*27*	1115
13	1204	*28*	1139
14	1365	*29*	1153
15	1192	*30*	496
		31	4

Confusion Matrix & Accuracy Output for XGBoost Model

Confusion Matrix and Statistics

Prediction	Reference: high	low	medium
high	310	52	163
low	305	9506	1940
medium	559	726	1245

Overall Statistics

Accuracy	0.7471
95% CI	(0.74, 0.754)
No Information Rate	0.6946
P-Value [Acc > NIR]	< 2.2e-16
Kappa	0.379
Mcnemar's Test P-Value	< 2.2e-16

Statistics by Class:

	Class: high	Class: low	Class: medium
Sensitivity	0.26405	0.9243	0.37186
Specificity	0.98423	0.5035	0.88785
Pos Pred Value	0.59048	0.8090	0.49209
Neg Pred Value	0.93950	0.7453	0.82869
Prevalence	0.07929	0.6946	0.22612
Detection Rate	0.02094	0.6420	0.08409
Detection Prevalence	0.03546	0.7937	0.17088
Balanced Accuracy	0.62414	0.7139	0.62986

Figure 6: XGBoost Confusion Matrix (Caret Output)

Confusion Matrix & Accuracy Output for Random Forest Model

```
Confusion Matrix and Statistics

          Reference
Prediction high  low medium
    high    302   57    196
    low     366 9499   2028
    medium  506  728   1124

Overall Statistics

               Accuracy : 0.7379
                 95% CI : (0.7307, 0.7449)
    No Information Rate : 0.6946
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3497
 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: high Class: low Class: medium
Sensitivity              0.25724     0.9237        0.33572
Specificity              0.98144     0.4706        0.89230
Pos Pred Value           0.54414     0.7987        0.47668
Neg Pred Value           0.93881     0.7305        0.82134
Prevalence               0.07929     0.6946        0.22612
Detection Rate           0.02040     0.6416        0.07592
Detection Prevalence     0.03748     0.8033        0.15926
Balanced Accuracy        0.61934     0.6971        0.61401
```

Figure 7: Random Forest Confusion Matrix (Caret Output)

Confusion Matrix & Accuracy Output for KNN Model

Confusion Matrix and Statistics

Prediction	Reference high	low	medium
high	47	15	33
low	797	9795	2690
medium	330	474	625

Overall Statistics

Accuracy	0.7069
95% CI	(0.6995, 0.7143)
No Information Rate	0.6946
P-Value [Acc > NIR]	0.000542
Kappa	0.1735
Mcnemar's Test P-Value	< 2.2e-16

Statistics by Class:

	Class: high	Class: low	Class: medium
Sensitivity	0.040034	0.9525	0.18668
Specificity	0.996479	0.2289	0.92983
Pos Pred Value	0.494737	0.7375	0.43737
Neg Pred Value	0.923391	0.6791	0.79644
Prevalence	0.079292	0.6946	0.22612
Detection Rate	0.003174	0.6616	0.04221
Detection Prevalence	0.006416	0.8971	0.09651
Balanced Accuracy	0.518256	0.5907	0.55825

Figure 8: KNN Confusion Matrix (Caret Output)

Confusion Matrix & Accuracy Output for LDA Model

	high	low	medium
high	181	105	170
low	741	9729	2602
medium	252	450	576

```
> mean(p_lda$class == v1$interest_level)

[1] 0.7082264
```

Figure 9: LDA Confusion Matrix

Confusion Matrix & Accuracy Output for QDA Model

	high	low	medium	
high	781	3302	1552	
low	15	1349	53	
medium	378	5633	1743	

> mean(p_qda$class == v1$interest_level) [1] 0.2615831

Figure 10: QDA Confusion Matrix