Liliane Giguere Samson #20112594

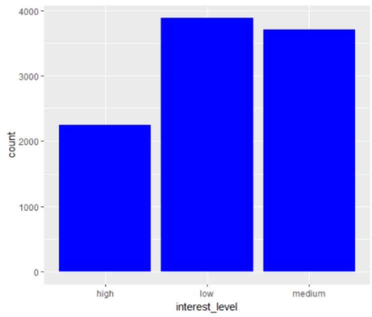
STAT 457 Project II

Introduction

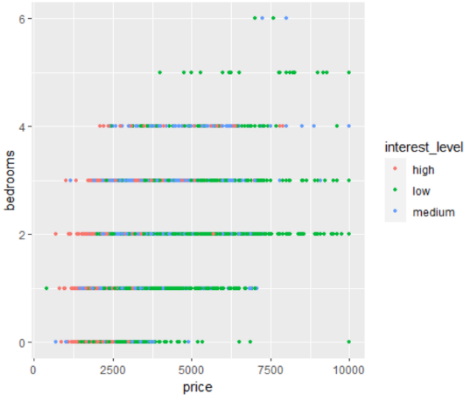
<u>Problem description</u>

The goal of this task is to try to predict the interest level of an apartment rental listing based on different variables such as number of bedrooms, price, location, and more. The interest level in this dataset is the number of inquiries a listing has in the duration that the listing was live on the rental site. It is important to note that this is a classification task; the target variable is categorical with 3 different categories of low, medium and high. We also want to maximize accuracy for this task.

Data exploration



This following plot shows the overall number of interest levels in the training dataset. There is almost double the amount of datapoints with low interest level than with high. There are also many medium level points, with it having the second highest count. This is important to note so that when we split the training dataset, we should ensure that both the new training and testing datasets keep this distribution of response variables.



Looking at the price variable, there is an extreme outlier where the price is 1,150,000. The majority of apartments where the price is more than 10,000 are classified as being of low interest. The following graph shows number of bedrooms vs price, with the target variable being showed by the color of the point. We can see that not many apartments have 5 or 6 bedrooms, and the ones that do are classified as low. Most of the high interest level listings seem to have a lower price, with the medium levels scattered throughout the plot. Clearly people want the lowest price per bedroom, so I created a variable that was the price/bathroom, and price/bedroom.

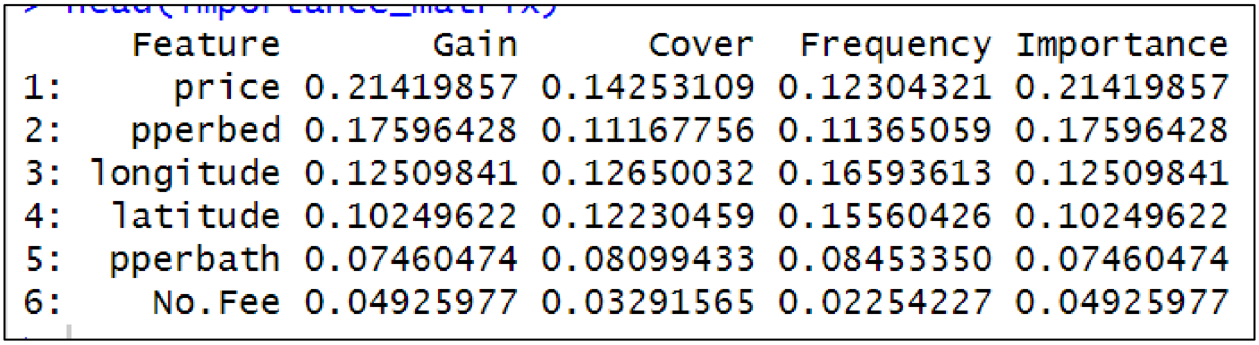
I also wanted to see if there were any apartments with 0 bedrooms and 0 bathrooms. There were 22 datapoints, with 20 being low, 1 being high and 1 being medium. Conceptually, these are strange points – how could an apartment have no bathroom and no bedroom. The test data set also has 11 points like this, so they will be left alone.

For the indicator variables, I took the mean of them and sorted by ascending order. “Eat.in.kitchen” had a mean of 0, meaning that every listing did not have this feature (had a value of 0). There seems to be a duplicate variable called “EAT.IN.KITCHEN” which has the true values. The next smallest means were “cable.satellite.tv”, “washer.dryer.in.unit”, “guarantors.accepted”, “wifi” and “concierge”. “Elevator” has a highest mean at 0.53 meaning almost half the rental listings have one. “Hardwood floors”, “cats allowed” and “dishwasher” are the next highest.

Exploratory data analysis

1. Which particular feature seems to have the most impact?

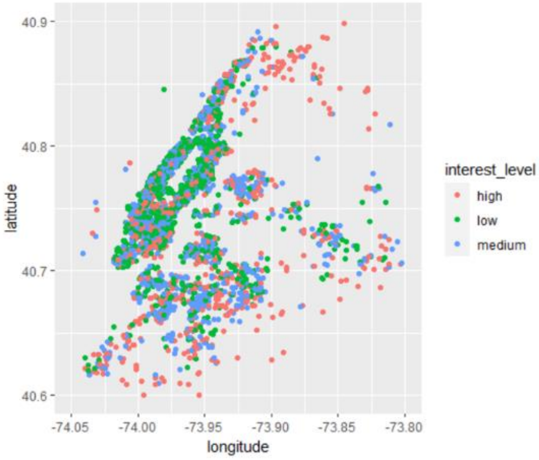
According to our xgboost model, price, price per bedroom, longitude, latitude, price per bathroom and no fee were all the most impactful.



```
> head(importance_matrix)
     Feature       Gain      Cover  Frequency Importance
1:     price 0.21419857 0.14253109 0.12304321 0.21419857
2:    pperbed 0.17596428 0.11167756 0.11365059 0.17596428
3: longitude 0.12509841 0.12650032 0.16593613 0.12509841
4:  latitude 0.10249622 0.12230459 0.15560426 0.10249622
5:  pperbath 0.07460474 0.08099433 0.08453350 0.07460474
6:    No.Fee 0.04925977 0.03291565 0.02254227 0.04925977
```

Price per bathroom, Dishwasher, Hardwood floors, Laundry in building and the total amount of “features” they have were the variables that correlated with interest level the most.

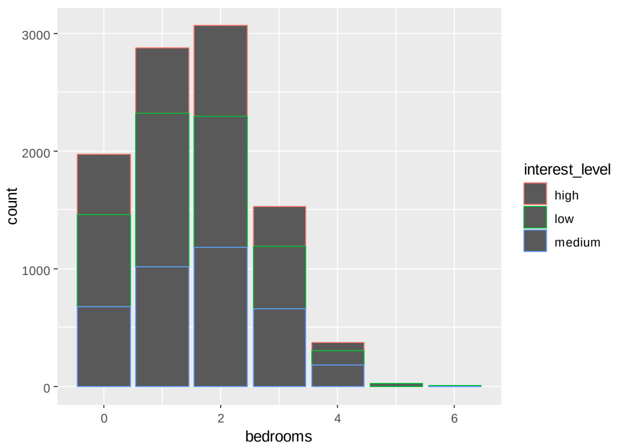
2. How does location affect the interest level?



We plotted latitude vs longitude to answer this question. We excluded 59 observations to narrow down the graph scope (we found one observation with 0 latitude and 0 longitude). Based on this graph, it seems that the low interest level points are much more tightly clustered together in the left side of the longitude axis, whereas the high interest level points are scattered throughout the plot, however mostly appear in the bottom half of the latitude plot. (Area representing Brooklyn) Brooklyn has a lot of high and medium interest levels, with not many low ones. There are also a couple of medium and high interest listings in New Jersey. Based off this plot, latitude and longitude seem to be good predictor variables to use in our models.

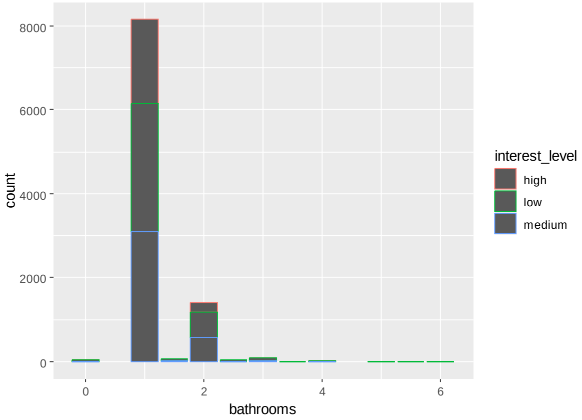
I therefore created a variable that used longitude and latitude that would classify each points to a neighbourhood of either: Uptown, Midtown, Downtown, Queens, The Bronx, Brooklyn, and then created dummy variables for each.

3. How does the number of bedrooms and bathrooms affect the interest level?



Most apartments have between 1-2 bedrooms. Proportionally, it seems that the number of bedrooms with the most interest is at 0. Low interest seems to dominate the 1 bedroom bar, with medium interest being the majority of the 2 bedroom bar. Medium interest is also the majoriy of the 3+ bedroom listings.

As for bathrooms, clearly most have 1. Interest level is mostly evenly split along this bar. The second most number of bathrooms that listings have is 2, and for these, low and medium interest are much more than the high interest. It seems that more people are interested in having one bathroom than 2, and that this variable may not be very impactful in our models.



Data cleaning

The data exploration stage showed us that this dataset requires some cleaning. First, the "Eat.in.kitchen" variable can be removed because it is a duplicate. The variables with extremely small means mentioned above will be removed to remove noise. They don't seem to be correlated with the interest level.

Next the datapoint with 0 longitude and 0 latitude was removed. I treated it as a "missing value" datapoint. The next one removed was the one with the price of 1,150,000. With a Google search, this is a condo listing, not a rental listing, and it is an extremely large outlier. No datapoint in the test dataset has a price of anywhere close to this value.

To see if we can eliminate any more predictor variables, I checked the correlation between each pair for all variables. I found another duplicate variable of "private.bathroom" and "Private.bathroom" that had a correlation of 1, so I removed the lowercase one. "Assigned parking space" and "private parking" also have a correlation of 1, meaning they are essentially duplicates here. I removed assigned parking space. "ft.doorman" has a correlation of 1 with "full.service.garage", "Washer.Dryer.in.building", "Outdoor.Entertainment.Space" and "Live.In.Superintendent". These are all essentially representing the same thing.

There are many columns in this dataset, making cross validation a very long process. Just looking at variable names, there are quite a few that seem to represent the same thing.

Variable 1	**Variable 2**	**outcome**
Childrens.Playroom	Children.s.Playroom	Get rid of both, each variable is mostly represented by lows, only 12 rows have a 1 for var1, only 6 row have a 1 for var2
Concierge	Concierge.service	Get rid of concierge service, only 6 rows have it
Full.time.doorman	F.T. doorman	Get rid of both, almost no apartments have this feature, and the ones that to are all low
Hardwood.floors	Hardwood.floors2	Get rid of var2, clearly a duplicate
Hardwood	Hardwood2	Get rid of var2, clearly a duplicate
High ceilings	High ceilings 1,2,3	Only keep high.ceiling.2 – has the most data
Gym.fitness, gym	Gym2, gym.in.building	Get rid of each, no more than 10 rows have this variable
Marble bath	Marble bathroom	Get rid of each, no more than 10 rows have this variable
Pre.war	Prewar, prewar	Get rid of var2
Roof.deck	Roof.deck2, ROOFDECK, Common.roof.deck	Get rid of all var2, clearly duplicates of roof.deck
Live.IN.SUPER	Live.in.superintendent, Live.In.Superintendent2, Live.in.Super2, Live.in.Super3	Get rid of all vars, clearly duplicates of var1
Valet	Valet.parking	Get rid of both, no more than 12 rows have these variables

Variables with a correlation of >0.9 (absolute value) are the following:

Variable 1	**Variable 2**	**Correlation**
Longitude	Latitude	-0.989
Cats allowed	Dogs allowed	0.925
prewar	Lowrise	0.951
prewar	Simplex	0.956
prewar	Hardwood	0.951
Simplex	Lowrise	0.918
Simplex	Hardwood	0.996
Lowrise	Hardwood	0.913
Marble bath	Granite Kitchen	0.913

The variable of street address is tricky to use. It has high cardinality, and there are many different formats of addresses for each variable. One will use St. and another with use street, one will have 455 as a number and the other 466-34 (because of an apartment or an attached house). I tried to play around with this variable a lot but string splitting based on space, or trying to use the grep function to character match, but overall I decided that including street address would make my model too complicated so I decided to leave out.

I will now run PCA analysis on these highly correlated variables to see if we can reduce dimensionality (except for latitude and longitude).

PCA analysis

This dataset has 169 features, with the majority being indicator variables for a certain condition such as is there laundry in the building. Therefore, PCA would be a valuable process to run to see if we can have less attributes and shorten the model complexity.

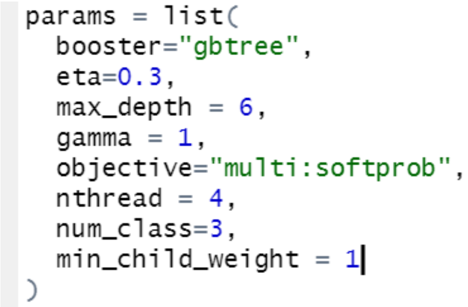
PCA analysis was conducted, however about 100 PCA components represented 90% of the variance. They will be used to see if using them is more helpful in our models.

ML Models

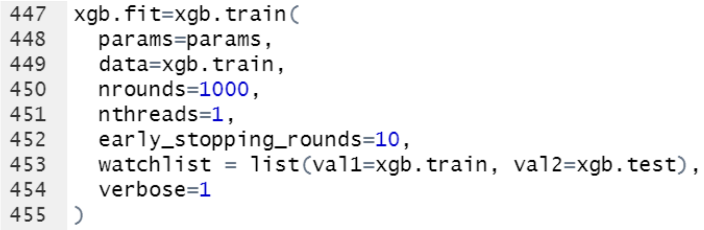
First, to just get an overall idea of the accuracy of our data and because of the simplicity to run it, I ran a logistic regression using the multinom() function in R. I achieved a baseline score of 0.998. Usually, logistic regression performs the poorest so I decided to move on to using a random forest.

I played around with this algorithm a lot, trying different features and tuning it to find the best parameters using the caret package. I found that the optimal score I could achieve was 0.87769, where mtry =24 and ntrees = 750. I attempted to use the random forests with the PCA components, and got a much worse score of 1.1, so decided that the data was better to use than the components.

The final model which provided the best score was xgboost. Xgboost needs all variables to be numerical, so the neighbourhood variable was removed. It also requires that the targest variable start at 0 and not 1, so 1 was subtracted from interest_level. I partitioned the train and test dataset from the original train so that the proportion of interest level was the tame in the train and test using createDataPartition. I initially used the following parameters and then trained the model.

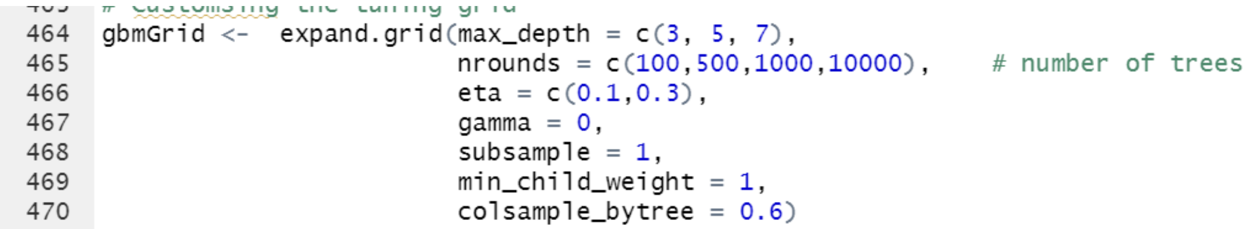


```
params = list(
  booster="gbtree",
  eta=0.3,
  max_depth = 6,
  gamma = 1,
  objective="multi:softprob",
  nthread = 4,
  num_class=3,
  min_child_weight = 1
)
```



```
xgb.fit=xgb.train(
  params=params,
  data=xgb.train,
  nrounds=1000,
  nthreads=1,
  early_stopping_rounds=10,
  watchlist = list(val1=xgb.train, val2=xgb.test),
  verbose=1
)
```

This model gave me a score of 8.5373. Now can we use cross validation to tune the parameters and improve the score. We tuned the parameters using a grid search with the following options and 5 fold cross validation.

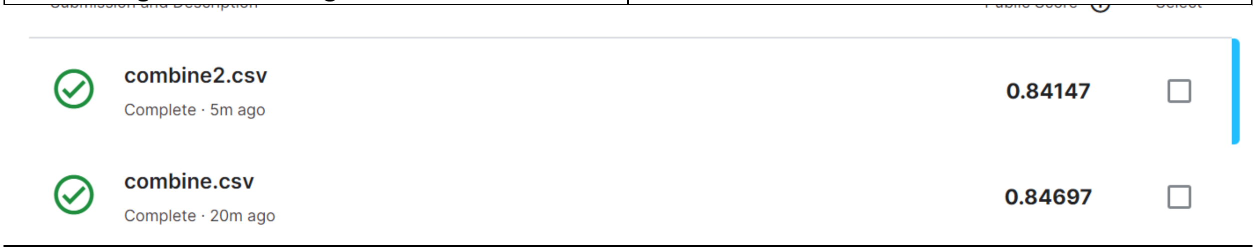


```
# customising the tuning grid
gbmGrid <-  expand.grid(max_depth = c(3, 5, 7),
                        nrounds = c(100,500,1000,10000),    # number of trees
                        eta = c(0.1,0.3),
                        gamma = 0,
                        subsample = 1,
                        min_child_weight = 1,
                        colsample_bytree = 0.6)
```

The best result was with eat=0.1, max_dept=3 and nrounds=100. This improved the score to 0.84989. I then averaged out the result of both xgboost models, and got my best score of 0.84697.

I then took the average of the best performing random forest, the best performing xgboost model, and the average of both that I had previously gotten (I did all the computation on excel). This led me to my best score of 0.84147.

Model	Score
Logistic regression	0.998
Random forest	0.87796
Xgboost	0.84697
Average of random forest results, xgboost results, and average results of 2 xgboost models	0.84147



Submission and Description | Public Score | Select

combine2.csv
Complete · 5m ago
0.84147 □

combine.csv
Complete · 20m ago
0.84697 □