Digital analysis of USA housing price and type

DNSC 6279 | Spring 2020

Dr. Zhengling Qi

Apr 27th, 2020

Group member: Yangzhuopeng Yi
yyi60gwu.edu

Shengqi Zhou
zsq947@gwu.edu

Yichen Li
liyichen_15@gwu.edu

Fengshu Xu
fxu8@gwu.edu

Contents

1.Abstract

The rental price of a house involves many aspects, which also decide by the value of itself; what's more, since houses are sorted by types, different house types, including apartment, condo, loft,etc., also could be regarded as a standard to determine the house rental price.
To figure out the relationship between all of these, some process of classification, several methods were applied in this report, such as KNN, SVM, random tree model. Finally, based on this dataset, the best approach from the machine learning methods was defined as *Random Forest*. In other words, based on the dataset, *Random Forest* could tell more things about house price *and* could play a vital role in further research and experiment.

2. Introduction

With the increase of population and widespread migration, more and more people care about how to spend less money to live in a comfortable and favorable house in recent years. Given that, analysts hope to find out which factors will impact on house rent as well as the relationship between house type and other factors, so that suggestion can be proposed for both tenants and landlords.
Based on that, the "USA Housing Listings" dataset is considered as a good dataset.It is the original data source from Craigslist, which is the world's largest collection of privately sold housing options, and contains enough information for analysis. Through the analysis of USA housing market data, we hope to find criteria that identically predicts housing sale prices and values, in order to give suggestions. The dataset is from Kaggle

3. Data Description and preprocessing

Data description

The "USA Housing Listings" dataset includes 22 variables (columns) with 384977 observations(rows). In detail, all observations are collected from all States across America, and all variables can be divided into two categories:

1. Numerical: rent per month, total square footage, latitude, and longitude
2. Categorical: number of beds, number of bathrooms, house region, house type, states, cats allowed, dogs allowed, smoking allowed, wheelchair access allowed, electric

vehicle charger, comes with furniture. In addition, we have all these variables' original link, description and id as verification.

Here is a part of the data summary:

```
     price                       type              sqfeet              beds        
 Min.   :0.000e+00   apartment    :186097   Min.   :      0   Min.   :   0.000  
 1st Qu.:8.190e+02   house        : 22219   1st Qu.:    750   1st Qu.:   1.000  
 Median :1.059e+03   townhouse    : 12869   Median :    950   Median :   2.000  
 Mean   :1.351e+04   condo        :  4711   Mean   :   1105   Mean   :   1.928  
 3rd Qu.:1.464e+03   duplex       :  4490   3rd Qu.:   1154   3rd Qu.:   2.000  
 Max.   :2.768e+09   manufactured:  3820   Max.   :8388607   Max.   :1100.000  
                     (Other)      :  1764                                       
     baths         cats_allowed     dogs_allowed    smoking_allowed wheelchair_access
 Min.   : 0.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000   Min.   :0.0000  
 1st Qu.: 1.000   1st Qu.:1.0000   1st Qu.:1.0000   1st Qu.:0.000   1st Qu.:0.0000  
 Median : 1.000   Median :1.0000   Median :1.0000   Median :1.000   Median :0.0000  
 Mean   : 1.479   Mean   :0.7793   Mean   :0.7532   Mean   :0.643   Mean   :0.1042  
 3rd Qu.: 2.000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:0.0000  
 Max.   :75.000   Max.   :1.0000   Max.   :1.0000   Max.   :1.000   Max.   :1.0000  

 electric_vehicle_charge comes_furnished               laundry_options  
 Min.   :0.00000         Min.   :0.00000                       :     0  
 1st Qu.:0.00000         1st Qu.:0.00000   laundry in bldg    : 31409  
 Median :0.00000         Median :0.00000   laundry on site    : 44419  
 Mean   :0.01755         Mean   :0.05822   no laundry on site:  3355  
 3rd Qu.:0.00000         3rd Qu.:0.00000   w/d hookups        : 54485  
 Max.   :1.00000         Max.   :1.00000   w/d in unit        :102302  

           parking_options        lat             long              state       
 off-street parking:125105   Min.   :-43.53   Min.   :-163.89   ca      : 24175  
 attached garage   : 38670   1st Qu.: 33.96   1st Qu.:-105.07   tx      : 15542  
 carport           : 38478   Median : 38.59   Median : -89.40   fl      : 15232  
 detached garage   : 16356   Mean   : 37.89   Mean   : -94.22   mi      :  9834  
 street parking    : 15362   3rd Qu.: 41.74   3rd Qu.: -81.57   oh      :  9246  
 no parking        :  1857   Max.   : 64.99   Max.   : 172.63   nc      :  8886  
 (Other)           :   142                                      (Other):153055  
```

Figure 1: variables description

From Figure1, the dataset has some unreasonable values, such as a house with 1100 beds or a house with 75 bathrooms. Therefore, cleaning the data is necessary.

Data preprocessing

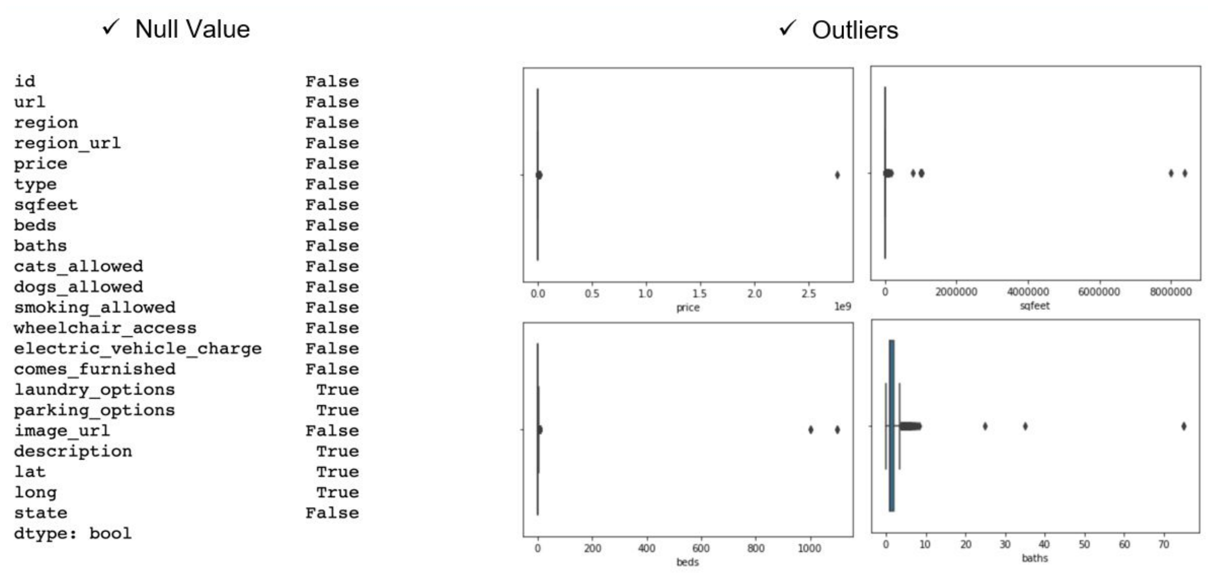

Figure 2: date remove outliers

The first chose to use the concept of interquartile to define outliers. More detailedly, for price and area, the value less than 1/10 1st Qu or more than 10*3rd Qu will be regarded as "outliers", and for beds and bathrooms, the value more than 10*3rd Qu will be regarded as “outliers” (we think no bed or no bathroom is acceptable.) The total number of outliers selected by this way is less than 5% of the whole dataset. Our group will explore how outliers will influence our model in the following parts. Also, we removed housing types with small amounts of data, since it’s hard to split them into a training and test dataset.

4. Methods

Our group tries to use regression and classification methods to solve two problems. The first problem is how to predict housing rent based on other features. For this problem, our group tries to build a regression model to predict the price. Before building the regression model, we need to determine which approach should we use to fit the model. There are three choices : least-square, ridge regression, and LASSO. In a dataset, if the number of features p is larger than the number of samples n, the least-squares regression coefficients are highly variable because some of variables are highly correlated with each other. In this case, we will consider using ridge regression or LASSO. However, in this dataset, the number of features is much less than the number of samples, so there is no need to use ridge regression or LASSO. Therefore, we chose the least square to fit our regression model. Then, we try to determine how many features should be included in our regression model. We can use variable pre-selection methods or forward stepwise selection. For this dataset, the first one will be better since forward stepwise selection isn't guaranteed to give us the best model. The best model chosen by forward stepwise selection with n variables may not contain every variable that is the best model with n-1 variables. Besides, the housing dataset only has 22 variables. Therefore, it is acceptable to consider every possibility. Actually, it only takes about ten seconds for R to get the best combo of variables. Next, we think about what kinds of regression models we should use. Our group considered the linear regression model and KNN. Here, it is hard to choose which model will be better only based on the concept. Thus, we used both two methods to fit the data, and we found the linear regression model has the lower test error, so we chose that to solve our first problem.

The second problem is how to classify the type of housing. For solving this problem, our group builds a classification model. We also consider many methods to build models, which include LDA, QDA, KNN, SVM, Classification Tree and Random Forest. We first exclude LDA and QDA from our choices, since the distribution of most of the variables in this dataset is not gaussian, which does not satisfy LDA and QDA assumptions. Then we tested KNN, SVM, Classification Tree, and Random Forest. We found that, for predicting the type of housing, Random Forest has higher accuracy than other models, so we chose Random Forest as our final model.

For the first problems, the variables listed here are what variable pre-selection methods chose for us. The last two interactions between variables are not in the original data set. It is what we added, and the pre-selection method thinks these two also are good predictors. The right side is the result of our linear regression model. We can see that the p-values for almost all variables and the whole model are very small, which means they are significant. The mean square test error is only around 4. It is very small compared with the housing rent, which means the model is relatively accurate.

And here is the KNN classification results for the housing type. The left part shows the final value used for the model is k=6. And the right part is the confusion matrix, we can see the balanced accuracy for all types are more than 50%, which means the model is relatively accurate.

5. Results

To find a prediction method for the housing rent, we first use the variable pre-selection method to find good predictors among all variables. In order to make our model more accurate, our group also adds two interaction variables (sqfeet:beds and sqfeet:baths) to the candidate variables. The following is the best price-predictors combo provided by variable pre-selection method:

type+sqfeet+beds+electric_vehicle_charge+lat+long+laundry_options+parking_options+sqfeet:beds+sqfeet:baths+smoking_allowed

Here is the linear regression result (only use training set)

```
Call:
lm(formula = price ~ type + sqfeet + beds + electric_vehicle_charge +
    lat + long + laundry_options + parking_options + sqfeet:beds +
    sqfeet:baths + smoking_allowed, data = training)

Residuals:
    Min      1Q  Median      3Q     Max
-4908.4  -281.4   -81.7   173.7 11422.4

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     8.295e+02  2.362e+01  35.122  < 2e-16 ***
typecondo                       1.931e+02  1.446e+01  13.360  < 2e-16 ***
typeduplex                     -1.119e+02  1.501e+01  -7.455 9.10e-14 ***
typehouse                      -1.408e+02  8.692e+00 -16.199  < 2e-16 ***
typemanufactured               -2.187e+02  1.655e+01 -13.216  < 2e-16 ***
typetownhouse                  -5.977e+01  9.320e+00  -6.413 1.43e-10 ***
sqfeet                          2.406e-01  1.266e-02  19.002  < 2e-16 ***
beds                           -3.639e+01  4.723e+00  -7.705 1.32e-14 ***
electric_vehicle_charge         4.189e+02  1.558e+01  26.884  < 2e-16 ***
lat                            -7.567e+00  3.877e-01 -19.520  < 2e-16 ***
long                           -6.047e+00  1.356e-01 -44.598  < 2e-16 ***
laundry_optionslaundry on site -6.646e+01  7.545e+00  -8.808  < 2e-16 ***
laundry_optionsno laundry on site -6.047e+01  1.865e+01  -3.243  0.00118 **
laundry_optionsw/d hookups     -1.314e+02  7.563e+00 -17.371  < 2e-16 ***
laundry_optionsw/d in unit      1.924e+02  6.702e+00  28.704  < 2e-16 ***
parking_optionscarport         -2.238e+02  7.747e+00 -28.893  < 2e-16 ***
parking_optionsdetached garage -1.296e+02  9.239e+00 -14.022  < 2e-16 ***
parking_optionsno parking      -1.046e+02  2.401e+01  -4.355 1.33e-05 ***
parking_optionsoff-street parking -2.761e+02  6.551e+00 -42.141  < 2e-16 ***
parking_optionsstreet parking  -1.858e+02  9.830e+00 -18.902  < 2e-16 ***
parking_optionsvalet parking    7.461e+02  8.145e+01   9.161  < 2e-16 ***
smoking_allowed                -1.088e+02  4.430e+00 -24.571  < 2e-16 ***
sqfeet:beds                     5.538e-02  3.742e-03  14.799  < 2e-16 ***
sqfeet:baths                    3.990e-02  3.390e-03  11.769  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 532.6 on 69590 degrees of freedom
Multiple R-squared:  0.3203,    Adjusted R-squared:  0.3201
F-statistic:  1426 on 23 and 69590 DF,  p-value: < 2.2e-16
```

Figure 3: date remove outliers

From the result, we can see the p-values of the whole model as well as all variables are very small, which means the whole model and all variables included in this model are significant.

Then our group estimates this model by calculating mean squared error using the test set, the mean squared error we got is 0.1385, which is very small related to the housing rate. It means the price production by our model is very accurate.

The following two pictures are our housing type prediction model using random forest. We predict housing type by the following predictors:
sqfeet+price+beds+baths+cats_allowed+dogs_allowed+smoking_allowed+wheelchair_access +electric_vehicle_charge+comes_furnished+laundry_options+parking_options

Figure 4 shows the training confusion matrix and Figure shows the testing confusion matrix. From figure 5 we can see the balanced accuracy for most housing types are higher than 80%, which means our type-prediction model is also relatively accurate.

```
Call:
 randomForest(formula = type ~ sqfeet + price + beds + baths +      cats_allowed + dogs_allowed
+ smoking_allowed + wheelchair_access +      electric_vehicle_charge + comes_furnished +
laundry_options +      parking_options, data = training3, mtry = 6, importance = TRUE,
ntree = 1000)
               Type of random forest: classification
                     Number of trees: 1000
No. of variables tried at each split: 6

        OOB estimate of  error rate: 8.66%
Confusion matrix:
             apartment condo duplex house manufactured townhouse class.error
apartment       126170   285    193  1564          153       764  0.02291507
condo             1670  1115     22   276           11       157  0.65702861
duplex            1331    36    899   702           19       126  0.71121105
house             2799    98    188 11834           72       399  0.23105913
manufactured       505    13     10   147         1909        15  0.26548673
townhouse         1648    70     63   724           11      6433  0.28114873
```

Figure 4: random forest model training confusion matrix

```
Confusion Matrix and Statistics

              Reference
Prediction     apartment condo duplex house manufactured townhouse
  apartment        53948   712    565  1196          219       749
  condo              120   467     14    51            1        37
  duplex             101    23    430    86            5        25
  house              753   112    268  5060           71       346
  manufactured        76     3     11    34          810         7
  townhouse          342    75     46   168            7      2670

Overall Statistics

               Accuracy : 0.9106
                 95% CI : (0.9085, 0.9127)
    No Information Rate : 0.795
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.731

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

Statistics by Class:

                     Class: apartment Class: condo Class: duplex Class: house
Sensitivity                    0.9748     0.335489      0.322339      0.76725
Specificity                    0.7588     0.996731      0.996485      0.97540
Pos Pred Value                 0.9400     0.676812      0.641791      0.76551
Neg Pred Value                 0.8861     0.986578      0.986887      0.97563
Prevalence                     0.7950     0.019998      0.019164      0.09474
Detection Rate                 0.7750     0.006709      0.006177      0.07269
Detection Prevalence           0.8245     0.009913      0.009625      0.09496
Balanced Accuracy              0.8668     0.666110      0.659412      0.87132
                     Class: manufactured Class: townhouse
Sensitivity                      0.72776          0.69640
Specificity                      0.99809          0.99030
Pos Pred Value                   0.86079          0.80713
Neg Pred Value                   0.99559          0.98244
Prevalence                       0.01599          0.05508
Detection Rate                   0.01164          0.03836
Detection Prevalence             0.01352          0.04752
Balanced Accuracy                0.86293          0.84335
```

Figure 5: random forest model tesing consusion matrix

6. Conclusion and Remarks

Based on the output of the regression model, we can find out how these selected variables will affect the rent of houses.

First, we check the effect from the type of housing. From our result, we can see the type, townhouse, has the biggest negative effect on the rent. On the contrary, the condo has the biggest positive effect on the rent. If an investor has spare money and wants to invest in the real estate industry, we will suggest him give priority to buying a condo. Then, the duplex will be the backup choice. The townhouse is the worst choice based on the result from the regression model.

After analyzing the type of house, we also can see some effects from the infrastructure on rent. The first one is laundry. Based on the result, whether laundry on site or no laundry on site both have negative effects on the rent. Only having the washer and dryer in the unit has a positive effect. Thus, we strongly suggest that property owners prepare washers and dryers for their residents. It is because they can charge higher rent if they prepare these equipment. If they really can not make it happen, preparing washer and dryers connections in units is another selection. Even this one also brings negative influence on rent, but it is much less than providing laundry on site or no laundry on site.

About parking. From the output of our model, we found that almost all parking options will bring negative effects on the rent. Only the valet parking brings a positive effect on rent. Within these options, a house offers off-street parking is most likely to have low rent. It has the biggest negative influence on the houses' value. According to the output, even no parking option is better than off-street parking.

All in all, if you want to buy a house as an investment, we suggest you choose a condo with a washer and dryer in the unit and offering valet parking services.

7. Reference

(Link: https://www.kaggle.com/austinreese/usa-housing-listings).

8. Appendix

variable pre-selection process:

```
```{r}
regfit11=regsubsets(price~type+sqfeet+beds+baths+cats_allowed+dogs_allowed+smoking_allowed+wheelchair_access+electric_vehicle_charge+comes_furnished+laundry_options+parking_options+lat+long+sqfeet*baths+sqfeet*beds,data=training,nvmax=50) #the command returns the best model given the number of regressors included. The argument "nvmax" allows you specify the size of the largest model to fit.

summary(regfit11)
```
```

```
Subset selection object
Call: regsubsets.formula(price ~ type + sqfeet + beds + baths + cats_allowed +
    dogs_allowed + smoking_allowed + wheelchair_access + electric_vehicle_charge +
    comes_furnished + laundry_options + parking_options + lat +
    long + sqfeet * baths + sqfeet * beds, data = training, nvmax = 50)
28 Variables  (and intercept)
```

```
28 Variables  (and intercept)
                                  Forced in Forced out
typecondo                             FALSE      FALSE
typeduplex                            FALSE      FALSE
typehouse                             FALSE      FALSE
typemanufactured                      FALSE      FALSE
typetownhouse                         FALSE      FALSE
sqfeet                                FALSE      FALSE
beds                                  FALSE      FALSE
baths                                 FALSE      FALSE
cats_allowed                          FALSE      FALSE
dogs_allowed                          FALSE      FALSE
smoking_allowed                       FALSE      FALSE
wheelchair_access                     FALSE      FALSE
electric_vehicle_charge               FALSE      FALSE
comes_furnished                       FALSE      FALSE
laundry_optionslaundry on site        FALSE      FALSE
laundry_optionsno laundry on site     FALSE      FALSE
laundry_optionsw/d hookups            FALSE      FALSE
laundry_optionsw/d in unit            FALSE      FALSE
parking_optionscarport                FALSE      FALSE
parking_optionsdetached garage        FALSE      FALSE
parking_optionsno parking             FALSE      FALSE
parking_optionsoff-street parking     FALSE      FALSE
parking_optionsstreet parking         FALSE      FALSE
parking_optionsvalet parking          FALSE      FALSE
lat                                   FALSE      FALSE
long                                  FALSE      FALSE
sqfeet:baths                          FALSE      FALSE
sqfeet:beds                           FALSE      FALSE
1 subsets of each size up to 28
Selection Algorithm: exhaustive
```

linear regression mean squared error for test set:

```
test_pred1=predict(lm.fit1,newdata=testing)
mean(testing$price-test_pred1)^2

```

[1] 0.1385302
```
```

knn result for regression:

```
69614 samples
   10 predictor

Pre-processing: centered (23), scaled (23)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 62652, 62654, 62653, 62651, 62653, 62652, ...
Resampling results across tuning parameters:

  k   RMSE      Rsquared   MAE
   1  469.4659  0.5310472  177.9418
   2  431.0071  0.5733494  182.7010
   3  421.6672  0.5822227  188.3189
   4  419.0359  0.5834981  192.9046
   5  418.2524  0.5833397  197.1441
   6  419.1623  0.5804561  201.5038
   7  420.0274  0.5781369  205.3512
   8  422.4178  0.5730734  209.1678
   9  424.0667  0.5696184  212.0796
  10  424.7561  0.5680961  214.8921

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was k = 5.
```

Mean squared error for test set:

```
```{r}
test_pred2=predict(knn.fit,newdata=testing)
mean(testing$price-test_pred2)^2
```
```

```
[1] 89.69935
```

Classification tree result:

```
```{r}
tree.housing=tree(type ~sqfeet+price+beds+baths+cats_allowed+dogs_allowed+smoking_allowed+wheelch
air_access+electric_vehicle_charge+comes_furnished+laundry_options+parking_options,data=training3
)
summary(tree.housing)
```
```

```
Classification tree:
tree(formula = type ~ sqfeet + price + beds + baths + cats_allowed +
    dogs_allowed + smoking_allowed + wheelchair_access + electric_vehicle_charge +
    comes_furnished + laundry_options + parking_options, data = training3)
Variables actually used in tree construction:
[1] "beds"            "cats_allowed"    "baths"           "parking_options"
[5] "sqfeet"
Number of terminal nodes:  8
Residual mean deviance:  1.157 = 187900 / 162400
Misclassification error rate: 0.1673 = 27179 / 162431
```

KNN for classification result:

```
k-Nearest Neighbors

11606 samples
   11 predictor
    6 classes: 'apartment', 'condo', 'duplex', 'house', 'manufactured', 'townhouse'

Pre-processing: centered (19), scaled (19)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 10446, 10445, 10445, 10444, 10446, 10444, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   1  0.8269867  0.4998433
   2  0.8157002  0.4621589
   3  0.8325869  0.4845014
   4  0.8350860  0.4830126
   5  0.8377567  0.4797779
   6  0.8378432  0.4742463
   7  0.8375829  0.4680537
   8  0.8368098  0.4637537
   9  0.8368962  0.4553366
  10  0.8362914  0.4507974

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 6.
```

```
Confusion Matrix and Statistics

              Reference
Prediction      apartment  condo duplex  house manufactured townhouse
  apartment        158509   2874   2313   6631         1475      4752
  condo               822    601     64    219           14       149
  duplex              503     42    370    444           19       116
  house              2772    408    939  11200          290      1177
  manufactured        697     15     47    258         1473        78
  townhouse          2719    238    269   1034           69      5232

Overall Statistics

               Accuracy : 0.8494
                 95% CI : (0.8479, 0.8509)
    No Information Rate : 0.795
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5253

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

Statistics by Class:

                     Class: apartment Class: condo Class: duplex Class: house
Sensitivity                    0.9547     0.143849      0.092454      0.56606
Specificity                    0.5785     0.993804      0.994513      0.97045
Pos Pred Value                 0.8978     0.321562      0.247657      0.66722
Neg Pred Value                 0.7672     0.982717      0.982483      0.95529
Prevalence                     0.7950     0.020007      0.019164      0.09475
Detection Rate                 0.7590     0.002878      0.001772      0.05363
Detection Prevalence           0.8454     0.008950      0.007154      0.08038
Balanced Accuracy              0.7666     0.568826      0.543483      0.76825
                     Class: manufactured Class: townhouse
Sensitivity                     0.441018          0.45480
Specificity                     0.994671          0.97806
Pos Pred Value                  0.573598          0.54722
Neg Pred Value                  0.990948          0.96853
Prevalence                      0.015994          0.05509
Detection Rate                  0.007054          0.02505
Detection Prevalence            0.012297          0.04578
Balanced Accuracy               0.717845          0.71643
```

SVM for classification results:

```
Statistics by Class:

                     Class: apartment Class: condo Class: duplex Class: house
Sensitivity                    0.9516     0.122676      0.063920      0.54695
Specificity                    0.5321     0.993778      0.996439      0.96776
Pos Pred Value                 0.8875     0.287003      0.259615      0.63969
Neg Pred Value                 0.7393     0.982297      0.981978      0.95329
Prevalence                     0.7950     0.020006      0.019162      0.09475
Detection Rate                 0.7565     0.002454      0.001225      0.05182
Detection Prevalence           0.8524     0.008551      0.004718      0.08101
Balanced Accuracy              0.7419     0.558227      0.530180      0.75735
                     Class: manufactured Class: townhouse
Sensitivity                     0.387975          0.33493
Specificity                     0.993504          0.97648
Pos Pred Value                  0.492618          0.45355
Neg Pred Value                  0.990085          0.96181
Prevalence                      0.015996          0.05509
Detection Rate                  0.006206          0.01845
Detection Prevalence            0.012598          0.04068
Balanced Accuracy               0.690740          0.65570
```