Blight Prediction

Alicia Brown

May 7, 2016

Contents

The data

The data provided for this project include 311 Service Calls, Crime Incidents, Blight Violations and Permits for Demolition. The first 3 datasets provide the foundation for a Buildings dataset that consists of unique locations as well as the source of derived features used in the model creation like *Number of Crimes*, *Number of Blight Violations*, and *Number of Service Calls*. These datasets were downloaded from the course site, but are also available via capstone project repo on github. All of the data comes from Socrata powered Detroit Open Data Portal, https://data.detroitmi.gov/.

Files

- Blight Violations
- 311 Service Calls
- Crime Incidents
- Demolition Permits

Note - I downloaded this Detroit Demolition dataset to use rather than one of building permits provided by instructor since it was cleaner data and contained only the essential fields needed to label known blight locations.

Data preparation & Tools

The greatest challenge in the provided data from Socrata is within the Location column because it concatenates all of the fields used in the geocoding process, and when address fields are included, line breaks are entered into the field and cause havoc until they are removed from the data file. Before analysis of the data could be undertaken, all files were initially formatted using Excel PowerQuery for removal of aforementioned line breaks and standardization of the street number and addresses. Then the data was loaded into FME, an amazing ETL tool, to validate and standardize the geographic coordinates and create well formatted incident and unique building files. Exploratory analysis and model creation was performed in python notebooks and RStudio.

Buildings

In order to derive a building, within the incident files, the latitude and longitude coordinates were rounded to 3 decimal points and then each file was individually joined with the demolition data that also had its latitude and longitude coordinates rounded to the same number of decimal points. Where there was a match in each file, then that building record was also labeled as blight. Any incident record that lacked a coordinate was excluded from the final dataset.

Features

Features drive the creation of predicton models because it is in their diversity that differences can be discovered that explain why one given building may be more prone to becoming blightful than another. In one of the readings for the course, Spatial Characteristics of Housing Abandonment, Dr. Morckel surmises that housing abandonment is a result of 3 key conditions - market conditions, gentrification and physical neglect. For this project, we are mostly focusing on the data evidence of neglect.

The first features added to the building dataset include a count of total 311 calls, crime incidents and blight violation citations for a given building. No filtering was performed on any of the incident datasets, because I didn't want to presume that calls about infrastructure or non-violent crimes are not related to a geographic inclination towards neglect.

A second set of features were added from a Property Values dataset found on the Detroit Data Portal that included appraised and taxed values, sales price, tax status, and whether it had been improved at any point. It also included well formatted street address and latitude and longitude coordinates which helped to further reduce the overall building dataset since any buildings that lack features are not useful in generating prediction models.

Features that were not included in this project that would be interesting to add so that the model accounts for possible gentrification and economic conditions could include - * Building Permits - alterations and other types that may signal gentrification in progress in neighborhood * Zillow Zestimate for an address * American Community Survey annual estimates on income, mortgage and rental at Census tract block level

It would also be interesting to consider time as a factor and perhaps calculate incident counts at 90 day, 180 day and annual intervals before demolition.

Models

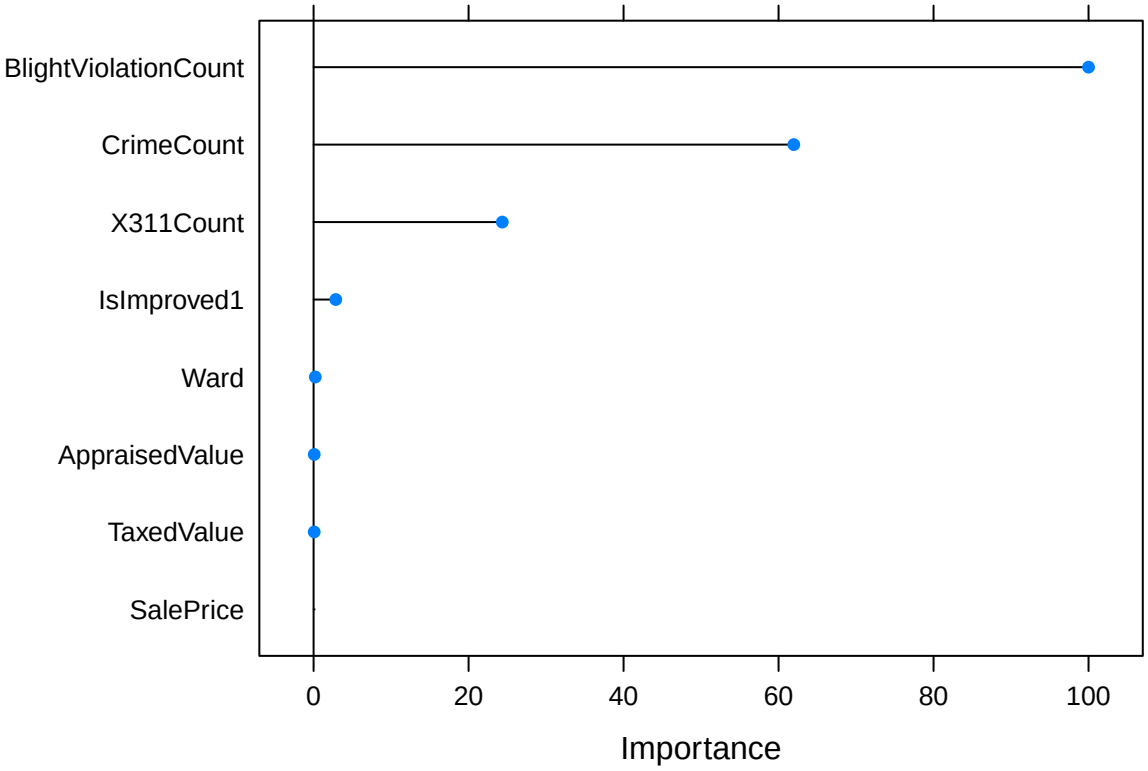
For the prediction of whether a location is blighted or not, I used logistic regression and classification tree models to answer that questions. Logistic regression models the probability that an outcome belongs to a category. Classification trees identify feature importance and provide a visual representation of the feature path taken to form the best performing model.

Training & Test datasets

In order to estimate the accuracy of a model, one must divide the data into training and test datasets. I have allocated 75% of the data for training the models.

Logistic Regression Model

```
##
## Call:
## NULL
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -1.321e-05  -7.600e-06  -7.100e-06  -6.910e-06   5.963e-04
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.967e+00  4.814e+02  -0.004    0.997
## CrimeCount            5.398e+01  1.189e+03   0.045    0.964
## X311Count             1.061e+01  5.945e+02   0.018    0.986
## BlightViolationCount  1.307e+02  1.784e+03   0.073    0.942
## Ward                 -4.988e-02  2.799e+02   0.000    1.000
## SalePrice            -4.728e-02  2.306e+03   0.000    1.000
## IsImproved1          -4.614e-01  2.180e+02  -0.002    0.998
## AppraisedValue       -1.735e+00  2.170e+04   0.000    1.000
## TaxedValue           -1.866e+00  2.450e+04   0.000    1.000
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3.4638e+04  on 120738  degrees of freedom
## Residual deviance: 1.3056e-05  on 120730  degrees of freedom
## AIC: 18
##
## Number of Fisher Scoring iterations: 25
```

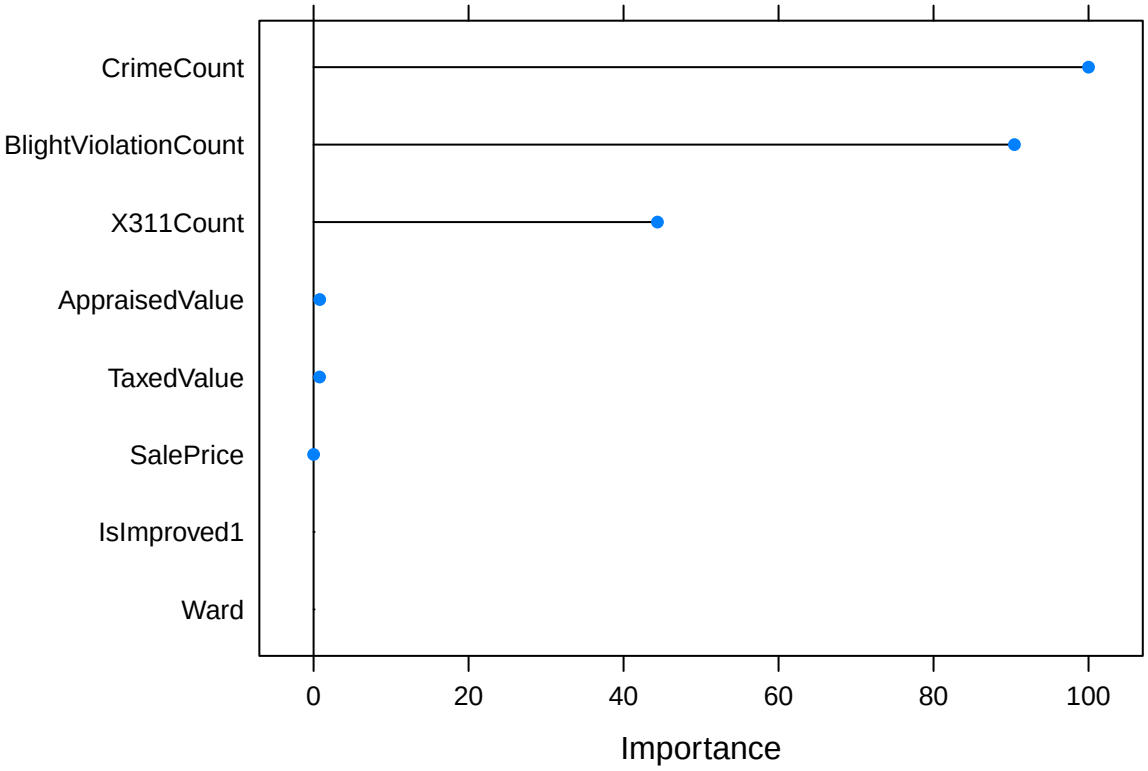



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 38937     0
##          1     0  1309
##
##                Accuracy : 1
##                  95% CI : (0.9999, 1)
##     No Information Rate : 0.9675
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.9675
##          Detection Rate : 0.9675
##    Detection Prevalence : 0.9675
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

Tree Model

The first model is a simple tree.

```
## CART
##
## 120739 samples
##      8 predictor
##      2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 120739, 120739, 120739, 120739, 120739, 120739, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa      Accuracy SD   Kappa SD
##   0.00814664  0.9998468  0.9975384  0.0001538805  0.002481588
##   0.11507128  0.9979073  0.9646532  0.0019522359  0.033220371
##   0.87678208  0.9812510  0.4468413  0.0144041125  0.474687025
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was cp = 0.00814664.

## n= 120739
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 120739 3928 0 (0.9674670156 0.0325329844)
##   2) BlightViolationCount< 0.5 117295  484 0 (0.9958736519 0.0041263481)
##     4) CrimeCount< 0.5 116843   32 0 (0.9997261282 0.0002738718) *
##     5) CrimeCount>=0.5 452    0 1 (0.0000000000 1.0000000000) *
##   3) BlightViolationCount>=0.5 3444    0 1 (0.0000000000 1.0000000000) *
```




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 38937     7
##          1     0  1302
##
##                Accuracy : 0.9998
##                  95% CI : (0.9996, 0.9999)
##     No Information Rate : 0.9675
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.9972
##  Mcnemar's Test P-Value : 0.02334
##
##             Sensitivity : 1.0000
##             Specificity : 0.9947
##          Pos Pred Value : 0.9998
##          Neg Pred Value : 1.0000
##              Prevalence : 0.9675
##          Detection Rate : 0.9675
##    Detection Prevalence : 0.9676
##       Balanced Accuracy : 0.9973
##
##        'Positive' Class : 0
##
```

Bagging Model

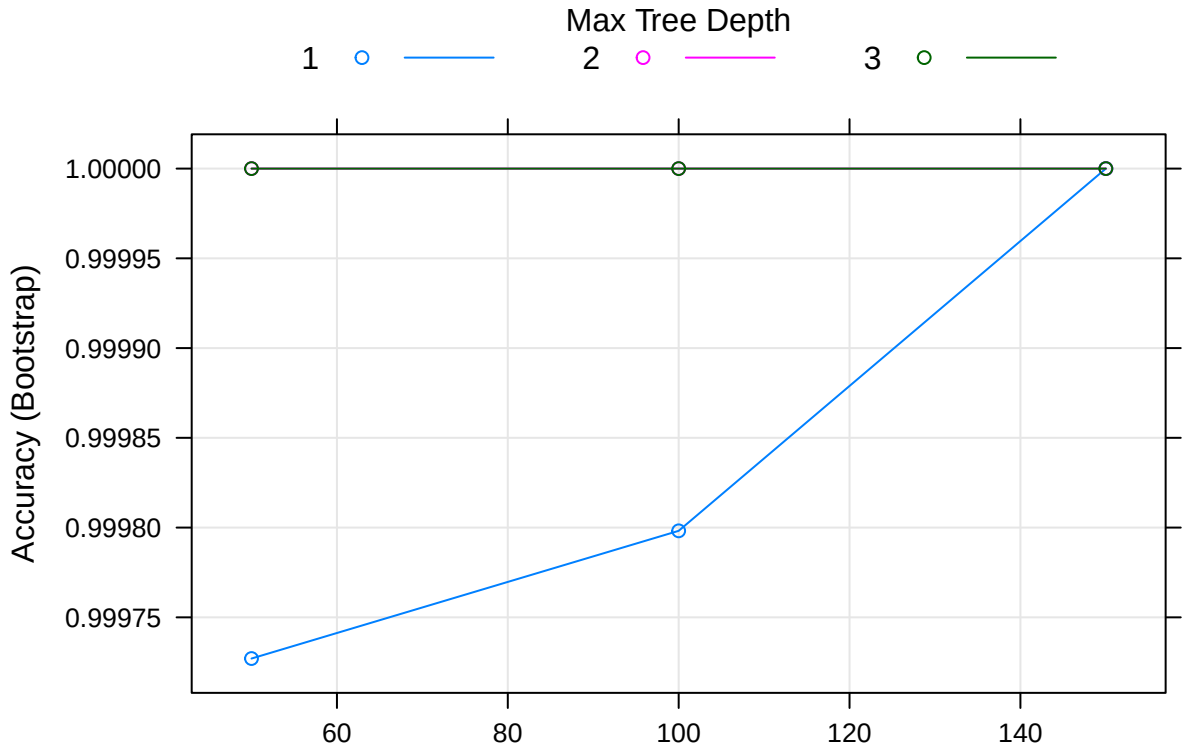
The next model involves Bootstrap Aggregating where the data is randomly resampled multiple times and the average is returned.

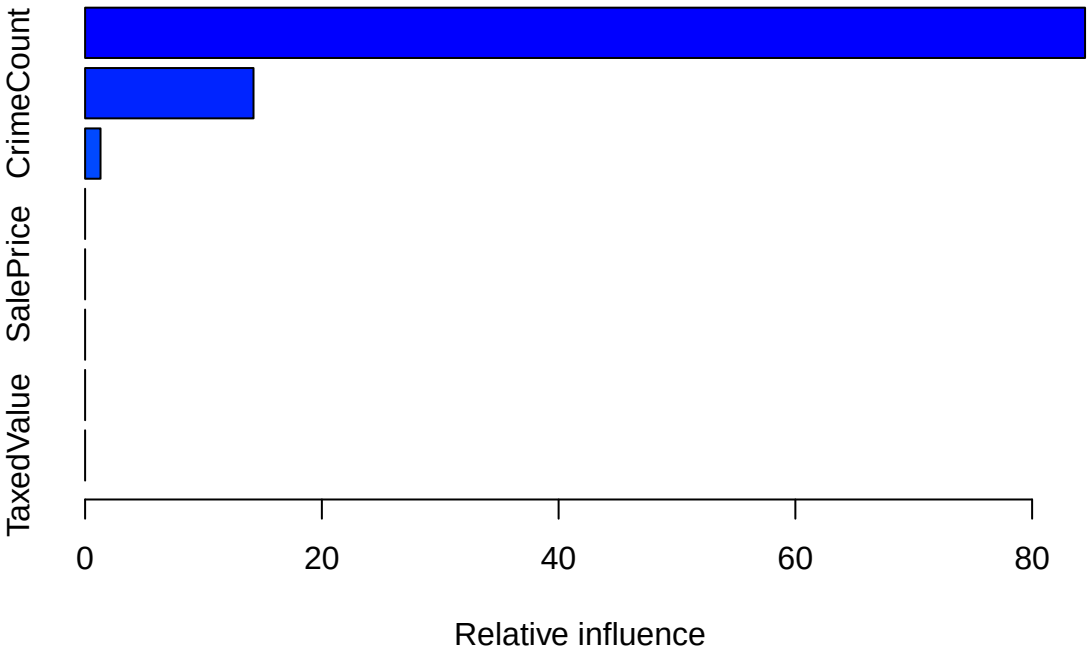
```
## Bagged CART
##
## 120739 samples
##      8 predictor
##      2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 120739, 120739, 120739, 120739, 120739, 120739, ...
## Resampling results
##
##   Accuracy  Kappa  Accuracy SD  Kappa SD
##   1         1      0            0
##
##

##
## Bagging classification trees with 25 bootstrap replications

## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 38937     0
##          1     0  1309
##
##                Accuracy : 1
##                  95% CI : (0.9999, 1)
##     No Information Rate : 0.9675
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.9675
##          Detection Rate : 0.9675
##    Detection Prevalence : 0.9675
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

Boosting Model

This model will combine weak classifiers so they can contribute to creating a more powerful model.

```
## Stochastic Gradient Boosting
##
## 120739 samples
##      8 predictor
##      2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 120739, 120739, 120739, 120739, 120739, 120739, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa      Accuracy SD
##   1                   50      0.9997270  0.9956461  6.350627e-05
##   1                  100      0.9997982  0.9967849  1.474613e-04
##   1                  150      1.0000000  1.0000000  0.000000e+00
##   2                   50      1.0000000  1.0000000  0.000000e+00
##   2                  100      1.0000000  1.0000000  0.000000e+00
##   2                  150      1.0000000  1.0000000  0.000000e+00
##   3                   50      1.0000000  1.0000000  0.000000e+00
##   3                  100      1.0000000  1.0000000  0.000000e+00
##   3                  150      1.0000000  1.0000000  0.000000e+00
##   Kappa SD
##   0.0009915973
##   0.0023451605
##   0.0000000000
##   0.0000000000
##   0.0000000000
##   0.0000000000
##   0.0000000000
##   0.0000000000
##   0.0000000000
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were n.trees = 50, interaction.depth
##  = 2, shrinkage = 0.1 and n.minobsinnode = 10.
```






```
##                                     var   rel.inf
## BlightViolationCount BlightViolationCount 84.470870
## CrimeCount                   CrimeCount 14.226556
## X311Count                     X311Count  1.302575
## Ward                               Ward  0.000000
## SalePrice                     SalePrice  0.000000
## IsImproved1                 IsImproved1  0.000000
```

```
## AppraisedValue             AppraisedValue  0.000000
## TaxedValue                     TaxedValue  0.000000
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 38937     0
##          1     0  1309
##
##                Accuracy : 1
##                  95% CI : (0.9999, 1)
##     No Information Rate : 0.9675
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.9675
##          Detection Rate : 0.9675
##    Detection Prevalence : 0.9675
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

Other Models

A Random Forest model is the most powerful of the tree classification models that involves bagging where it also resamples the feature combinations. However, the complexity of its algorithm causes long processing and in this case caused memory exceptions on 2 different and otherwise powerful macbookpro and imac computers, so was unable to run this model successfully.

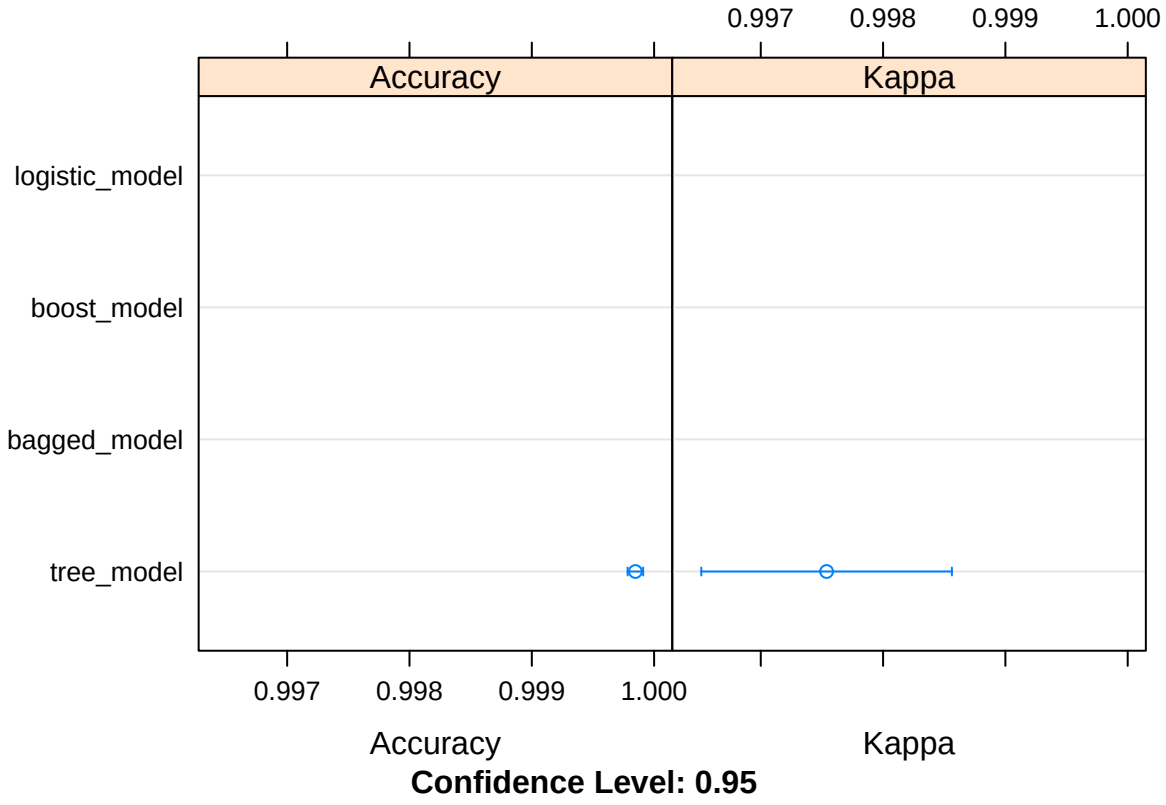
Model Comparison

```
# compare
results = resamples(list(tree_model = tree_model,
                         bagged_model = bagged_model,
                         boost_model = boost_model,
                         logistic_model = logistic_model))

# compare accuracy and kappa
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
```

```
##
## Models: tree_model, bagged_model, boost_model, logistic_model
## Number of resamples: 25
##
## Accuracy
##                   Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## tree_model     0.9996  0.9997 0.9998 0.9998       1    1    0
## bagged_model   1.0000  1.0000 1.0000 1.0000       1    1    0
## boost_model    1.0000  1.0000 1.0000 1.0000       1    1    0
## logistic_model 1.0000  1.0000 1.0000 1.0000       1    1    0
##
## Kappa
##                   Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## tree_model     0.9932  0.9953 0.9967 0.9975       1    1    0
## bagged_model   1.0000  1.0000 1.0000 1.0000       1    1    0
## boost_model    1.0000  1.0000 1.0000 1.0000       1    1    0
## logistic_model 1.0000  1.0000 1.0000 1.0000       1    1    0
```

```
# plot results
dotplot(results)
```



Conclusion

Tree model wins and suggests that crime and blight violations counts are key factors in whether a building is blight.