Image Segmentation -Capstone Project

Radhai

April 21, 2018

## Project Goal

The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.

Segmentation is generally the first stage in any attempt to analyze or interpret an image automatically. Segmentation bridges the gap between low-level image processing and high-level image processing.

This has application in various areas: Industrial inspection Optical character recognition (OCR) Tracking of objects in a sequence of images Classification of terrains visible in satellite images. Detection and measurement of bone, tissue, etc., in medical images.

## Dataset

I downloaded the data available in the following link for machine learning purpose: [Image Segmentation](https://archive.ics.uci.edu/ml/datasets/Image+Segmentation)

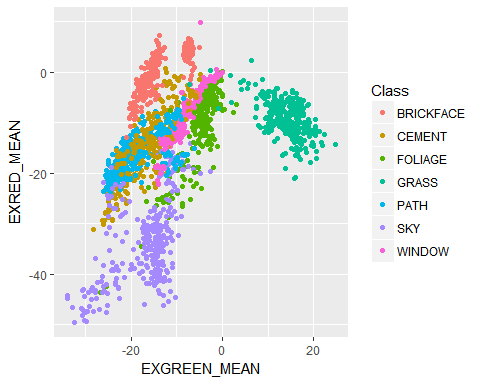
## My approach in classifying this data:

1. Data Cleaning The dataset does not have null values. Most of the field values are pre-processed and hence does not seem to need an extensive data cleaning. The classification field does not have a field name, the field names has “.” character. I need to do correct them.
2. Pre-processing: Use the R functions like: summary, cor, plot, lm, preprocessing commands in “caret” package to preprocess and understand the relationship between different columns and the Class variable.
3. Create a model that will classify the data accurately into seven classes.
4. Test the model on the test data. I tried with SVM (Support Vector Machine) algorithm, GLM, Tree and RandomForest models with cross validation technique for this classification problem.

## Exploratory Analysis

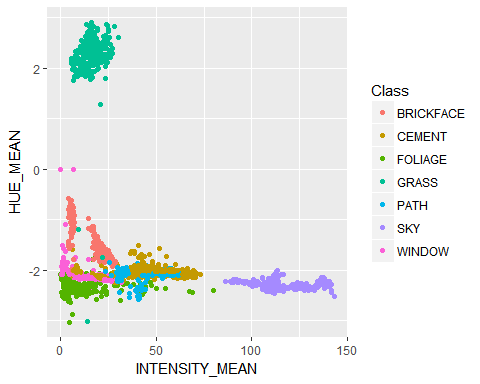
### EXGREEN vs EXRED

ggplot(imageSeg,aes(x=EXGREEN\_MEAN,y=EXRED\_MEAN,col=Class))+geom\_jitter()



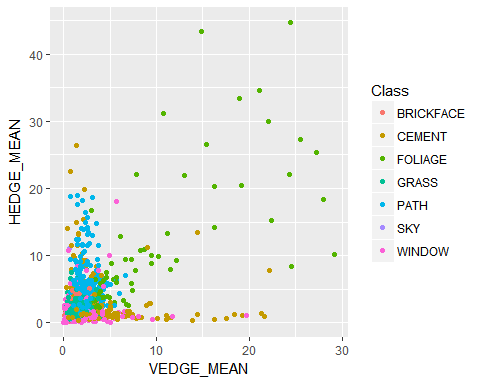
Obviously Grass has the highest EXGREEN\_MEAN and Brickface has high EXRED\_MEAN, as anyone would expect. In addition, we can see that Sky has low EXRED\_MEAN as well as EXGREEN\_MEAN.

### INTENSITY vs HUE



Evidently, we can see Grass has the greatest HUE\_MEAN, while FOLIAGE has the lowest HUE\_MEAN. Also, Sky has the greatest INTENSITY\_MEAN.

### VEDGE vs HEDGE

As per the plot, the variables are correlated as shown by the numbers above. 

Also we can observe that Brickface has very low HEDGE\_MEAN and VEDGE\_MEAN. Also Cement has very low HEDGE\_MEAN.

### Correlation between the plotted features

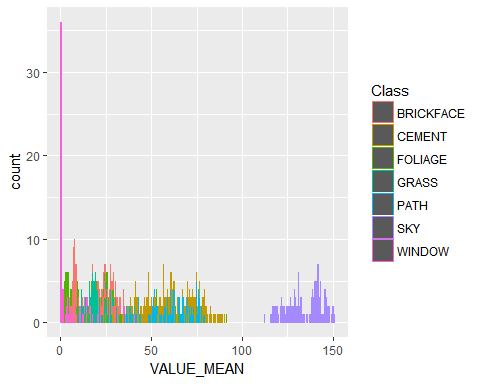
## [1] 0.435952

## [1] -0.3279644

## [1] 0.5422807

### A bar plot on VALUE\_MEAN for all classes

## Warning: position\_stack requires non-overlapping x intervals



This bar plot shows the high value for VALUE\_MEAN for Sky class though it does not have significant display for other classes.

## Summary of all the observed features of our dataset

## Class REGION\_CENTROID\_COL REGION\_CENTROID\_ROW  
## BRICKFACE:300 Min. : 1.0 Min. : 11.0   
## CEMENT :300 1st Qu.: 62.0 1st Qu.: 81.0   
## FOLIAGE :300 Median :121.0 Median :122.0   
## GRASS :300 Mean :124.9 Mean :123.5   
## PATH :300 3rd Qu.:188.2 3rd Qu.:171.2   
## SKY :300 Max. :254.0 Max. :251.0   
## WINDOW :300   
## REGION\_PIXEL\_COUNT SHORT\_LINE\_DENSITY\_5 SHORT\_LINE\_DENSITY\_2  
## Min. :9 Min. :0.00000 Min. :0.00000   
## 1st Qu.:9 1st Qu.:0.00000 1st Qu.:0.00000   
## Median :9 Median :0.00000 Median :0.00000   
## Mean :9 Mean :0.01492 Mean :0.00455   
## 3rd Qu.:9 3rd Qu.:0.00000 3rd Qu.:0.00000   
## Max. :9 Max. :0.33333 Max. :0.22222   
##   
## VEDGE\_MEAN VEDGE\_SD HEDGE\_MEAN   
## Min. : 0.0000 Min. : 0.0000 Min. : 0.0000   
## 1st Qu.: 0.7222 1st Qu.: 0.3496 1st Qu.: 0.8333   
## Median : 1.2778 Median : 0.8333 Median : 1.4444   
## Mean : 1.8908 Mean : 5.7083 Mean : 2.4068   
## 3rd Qu.: 2.2222 3rd Qu.: 1.8074 3rd Qu.: 2.5556   
## Max. :29.2222 Max. :991.7184 Max. :44.7222   
##   
## HEDGE\_SD INTENSITY\_MEAN RAWRED\_MEAN RAWBLUE\_MEAN   
## Min. : 0.0000 Min. : 0.000 Min. : 0.00 Min. : 0.000   
## 1st Qu.: 0.4216 1st Qu.: 7.472 1st Qu.: 7.00 1st Qu.: 9.667   
## Median : 0.9897 Median : 21.667 Median : 19.67 Median : 27.778   
## Mean : 7.9042 Mean : 37.048 Mean : 32.81 Mean : 44.206   
## 3rd Qu.: 2.2519 3rd Qu.: 53.278 3rd Qu.: 47.33 3rd Qu.: 65.000   
## Max. :1386.3292 Max. :143.444 Max. :137.11 Max. :150.889   
##   
## RAWGREEN\_MEAN EXRED\_MEAN EXBLUE\_MEAN EXGREEN\_MEAN   
## Min. : 0.000 Min. :-49.667 Min. :-12.444 Min. :-33.889   
## 1st Qu.: 6.222 1st Qu.:-18.583 1st Qu.: 4.306 1st Qu.:-17.000   
## Median : 20.444 Median :-10.889 Median : 19.667 Median :-11.000   
## Mean : 34.131 Mean :-12.723 Mean : 21.474 Mean : -8.751   
## 3rd Qu.: 46.389 3rd Qu.: -4.222 3rd Qu.: 36.111 3rd Qu.: -3.222   
## Max. :142.556 Max. : 9.889 Max. : 82.000 Max. : 24.667   
##   
## VALUE\_MEAN SATURATION\_MEAN HUE\_MEAN   
## Min. : 0.00 Min. :0.0000 Min. :-3.044   
## 1st Qu.: 11.78 1st Qu.:0.2849 1st Qu.:-2.189   
## Median : 28.67 Median :0.3751 Median :-2.053   
## Mean : 45.16 Mean :0.4273 Mean :-1.365   
## 3rd Qu.: 65.00 3rd Qu.:0.5402 3rd Qu.:-1.566   
## Max. :150.89 Max. :1.0000 Max. : 2.912   
##

As can be seen from the summary, most of the variable have huge difference between their min and the max values. Also, we can see the each variance are in scales that vary highly. As a common practice, we can scale and center the data so as to compare them in a better way.

## Preprocessing data

### a look on data before and after transformation

## REGION\_CENTROID\_COL REGION\_CENTROID\_ROW SHORT\_LINE\_DENSITY\_5  
## Min. :-1.70111 Min. :-1.95857 Min. :-0.3637   
## 1st Qu.:-0.86387 1st Qu.:-0.73972 1st Qu.:-0.3637   
## Median :-0.05408 Median :-0.02583 Median :-0.3637   
## Mean : 0.00000 Mean : 0.00000 Mean : 0.0000   
## 3rd Qu.: 0.86894 3rd Qu.: 0.83172 3rd Qu.:-0.3637   
## Max. : 1.77137 Max. : 2.22033 Max. : 7.7617   
## VEDGE\_MEAN VEDGE\_SD HEDGE\_MEAN HEDGE\_SD   
## Min. :-0.7137 Min. :-0.12688 Min. :-0.69360 Min. :-0.1478   
## 1st Qu.:-0.4411 1st Qu.:-0.11911 1st Qu.:-0.45345 1st Qu.:-0.1399   
## Median :-0.2314 Median :-0.10836 Median :-0.27733 Median :-0.1293   
## Mean : 0.0000 Mean : 0.00000 Mean : 0.00000 Mean : 0.0000   
## 3rd Qu.: 0.1251 3rd Qu.:-0.08671 3rd Qu.: 0.04288 3rd Qu.:-0.1057   
## Max. :10.3159 Max. :21.91652 Max. :12.19482 Max. :25.7789   
## INTENSITY\_MEAN RAWRED\_MEAN RAWBLUE\_MEAN RAWGREEN\_MEAN   
## Min. :-0.9715 Min. :-0.9375 Min. :-1.0160 Min. :-0.9401   
## 1st Qu.:-0.7755 1st Qu.:-0.7374 1st Qu.:-0.7938 1st Qu.:-0.7687   
## Median :-0.4033 Median :-0.3755 Median :-0.3776 Median :-0.3770   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.4256 3rd Qu.: 0.4151 3rd Qu.: 0.4779 3rd Qu.: 0.3377   
## Max. : 2.7900 Max. : 2.9806 Max. : 2.4519 Max. : 2.9866   
## EXRED\_MEAN EXBLUE\_MEAN EXGREEN\_MEAN VALUE\_MEAN   
## Min. :-3.1880 Min. :-1.72575 Min. :-2.1658 Min. :-1.0527   
## 1st Qu.:-0.5057 1st Qu.:-0.87351 1st Qu.:-0.7107 1st Qu.:-0.7782   
## Median : 0.1583 Median :-0.09194 Median :-0.1938 Median :-0.3845   
## Mean : 0.0000 Mean : 0.00000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.7336 3rd Qu.: 0.74475 3rd Qu.: 0.4763 3rd Qu.: 0.4624   
## Max. : 1.9513 Max. : 3.07957 Max. : 2.8791 Max. : 2.4645   
## SATURATION\_MEAN HUE\_MEAN   
## Min. :-1.8702 Min. :-1.0873   
## 1st Qu.:-0.6230 1st Qu.:-0.5332   
## Median :-0.2285 Median :-0.4452   
## Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.4945 3rd Qu.:-0.1299   
## Max. : 2.5070 Max. : 2.7700

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.0 62.0 121.0 124.9 188.2 254.0

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -1.70111 -0.86387 -0.05408 0.00000 0.86894 1.77137

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -49.667 -18.583 -10.889 -12.723 -4.222 9.889

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -3.1880 -0.5057 0.1583 0.0000 0.7336 1.9513

## Applying SVM (Support Vector Machine) algorithm

##   
## Call:  
## svm(formula = Class ~ ., data = imageSegTranscl)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: radial   
## cost: 1   
## gamma: 0.05882353   
##   
## Number of Support Vectors: 656  
##   
## ( 24 68 188 136 151 16 73 )  
##   
##   
## Number of Classes: 7   
##   
## Levels:   
## BRICKFACE CEMENT FOLIAGE GRASS PATH SKY WINDOW

The summary shows the default values assumed for each of the parameters used with this function.

By default, the classification type is chosen and radial kernel is used. The cost value is 1 and gamma is .06.

### Prediction using SVM model

## y  
## pred BRICKFACE CEMENT FOLIAGE GRASS PATH SKY WINDOW  
## BRICKFACE 295 0 0 0 0 0 1  
## CEMENT 0 285 4 0 0 0 14  
## FOLIAGE 0 0 284 1 0 0 49  
## GRASS 0 0 0 298 0 0 0  
## PATH 0 0 0 1 300 0 0  
## SKY 0 0 0 0 0 300 0  
## WINDOW 5 15 12 0 0 0 236

### Creating a tuned model

##   
## Call:  
## svm(formula = Class ~ ., data = imageSegTranscl, cost = 10, gamma = 0.5)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: radial   
## cost: 10   
## gamma: 0.5   
##   
## Number of Support Vectors: 699  
##   
## ( 61 71 147 173 143 46 58 )  
##   
##   
## Number of Classes: 7   
##   
## Levels:   
## BRICKFACE CEMENT FOLIAGE GRASS PATH SKY WINDOW

## y  
## pred2 BRICKFACE CEMENT FOLIAGE GRASS PATH SKY WINDOW  
## BRICKFACE 300 0 0 0 0 0 0  
## CEMENT 0 300 0 0 0 0 0  
## FOLIAGE 0 0 298 0 0 0 5  
## GRASS 0 0 0 300 0 0 0  
## PATH 0 0 0 0 300 0 0  
## SKY 0 0 0 0 0 300 0  
## WINDOW 0 0 2 0 0 0 295

### Validating the tuned model on test data

## y2  
## predtest BRICKFACE CEMENT FOLIAGE GRASS PATH SKY WINDOW  
## BRICKFACE 30 0 0 0 0 0 0  
## CEMENT 0 30 1 0 0 0 0  
## FOLIAGE 0 0 29 0 0 0 1  
## GRASS 0 0 0 30 0 0 0  
## PATH 0 0 0 0 30 0 0  
## SKY 0 0 0 0 0 30 0  
## WINDOW 0 0 0 0 0 0 29

# GLM

Since glm required the classification variable to be of numeric type, I have created a numeric variable with Hence I added a numeric column with values 1 through 7 corresponding to every class.

I also tried splitting my training data set into 2 sets, just to try with sample.split function.

### GLM model1

glmodel = glm(cl ~ .,data = traini)  
summary(glmodel)

##   
## Call:  
## glm(formula = cl ~ ., data = traini)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.2917 -0.7394 0.0305 0.7547 3.4913   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.579e+00 1.878e-01 8.411 < 2e-16 \*\*\*  
## REGION\_CENTROID\_COL 3.296e-03 4.176e-04 7.893 6.06e-15 \*\*\*  
## REGION\_CENTROID\_ROW 2.198e-02 7.542e-04 29.139 < 2e-16 \*\*\*  
## SHORT\_LINE\_DENSITY\_5 -8.033e-01 7.609e-01 -1.056 0.291285   
## VEDGE\_MEAN 1.976e-02 1.749e-02 1.129 0.259002   
## VEDGE\_SD -1.746e-03 1.107e-03 -1.577 0.114966   
## HEDGE\_MEAN 2.929e-02 1.543e-02 1.899 0.057843 .   
## HEDGE\_SD -3.096e-03 1.385e-03 -2.236 0.025515 \*   
## INTENSITY\_MEAN -2.985e+04 1.635e+04 -1.826 0.068053 .   
## RAWRED\_MEAN 2.178e+04 1.593e+04 1.367 0.171812   
## RAWBLUE\_MEAN -8.517e+03 1.297e+04 -0.657 0.511497   
## RAWGREEN\_MEAN 1.658e+04 1.623e+04 1.022 0.307104   
## EXRED\_MEAN -1.015e+04 2.284e+04 -0.444 0.656879   
## EXBLUE\_MEAN -4.979e+01 2.219e+04 -0.002 0.998210   
## EXGREEN\_MEAN -8.416e+03 2.299e+04 -0.366 0.714362   
## VALUE\_MEAN 1.250e-01 3.590e-02 3.482 0.000514 \*\*\*  
## SATURATION\_MEAN -2.541e+00 2.080e-01 -12.220 < 2e-16 \*\*\*  
## HUE\_MEAN -3.514e-01 5.746e-02 -6.115 1.26e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 1.242078)  
##   
## Null deviance: 5460.0 on 1364 degrees of freedom  
## Residual deviance: 1673.1 on 1347 degrees of freedom  
## AIC: 4189.5  
##   
## Number of Fisher Scoring iterations: 2

We can see the independent variables, REGION\_CENTROID\_COL, REGION\_CENTROID\_ROW, VALUE\_MEAN, SATURATION\_MEAN, and HUE\_MEAN as highly significant.

### GLM model2

Let try making a model with just the significant ones.

##   
## Call:  
## glm(formula = cl ~ REGION\_CENTROID\_COL + REGION\_CENTROID\_ROW +   
## VALUE\_MEAN + SATURATION\_MEAN + HUE\_MEAN + HEDGE\_MEAN + HEDGE\_SD +   
## INTENSITY\_MEAN, data = traini)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.4583 -0.5470 0.1640 0.7875 3.4462   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.1028668 0.1971075 5.595 2.66e-08 \*\*\*  
## REGION\_CENTROID\_COL 0.0038376 0.0004661 8.233 4.24e-16 \*\*\*  
## REGION\_CENTROID\_ROW 0.0210694 0.0008335 25.277 < 2e-16 \*\*\*  
## VALUE\_MEAN 0.0856509 0.0112028 7.645 3.92e-14 \*\*\*  
## SATURATION\_MEAN -0.7615768 0.2048730 -3.717 0.000210 \*\*\*  
## HUE\_MEAN 0.2469619 0.0299865 8.236 4.15e-16 \*\*\*  
## HEDGE\_MEAN 0.0624955 0.0164473 3.800 0.000151 \*\*\*  
## HEDGE\_SD -0.0023876 0.0014428 -1.655 0.098185 .   
## INTENSITY\_MEAN -0.0952022 0.0126790 -7.509 1.08e-13 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 1.570607)  
##   
## Null deviance: 5460.0 on 1364 degrees of freedom  
## Residual deviance: 2129.7 on 1356 degrees of freedom  
## AIC: 4500.9  
##   
## Number of Fisher Scoring iterations: 2

### Prediction of both the models on splitted test data

##   
## FALSE TRUE  
## 1 0 105  
## 2 1 104  
## 3 1 104  
## 4 0 105  
## 5 0 105  
## 6 0 105  
## 7 0 105

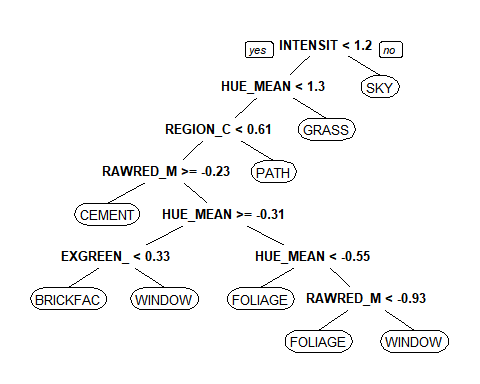
##   
## FALSE TRUE  
## 1 0 105  
## 2 1 104  
## 3 0 105  
## 4 0 105  
## 5 0 105  
## 6 0 105  
## 7 0 105

The prediction seems to be close to perfect. The one with the significant features can reduce one more mistake in the prediction. I also tested the model on the test data that was downloaded with this dataset.

##   
## FALSE TRUE  
## 1 0 30  
## 2 2 28  
## 3 0 30  
## 4 0 30  
## 5 0 30  
## 6 0 30  
## 7 0 30

The test data prediction has also show a very low error rate (2/210=.01).

# Tree model 1 without Cross Validation

 The tree has picked quite a set of independent variables, of which HUE\_MEAN and RAWRED\_MEAN has been used more than once.

### Prediction on the test data:

## predicttree  
## BRICKFACE CEMENT FOLIAGE GRASS PATH SKY WINDOW  
## BRICKFACE 29 0 0 0 0 0 1  
## CEMENT 0 24 1 0 2 0 3  
## FOLIAGE 1 3 15 0 0 0 11  
## GRASS 0 0 0 30 0 0 0  
## PATH 0 0 0 0 30 0 0  
## SKY 0 0 0 0 0 30 0  
## WINDOW 1 1 1 0 0 0 27

The prediction is not as good as the glm model.

# Tree model 2 with cross validation

We are choosing the number of folds for cross validation to be 10 and the range of cp parameter to be from .01 to .50

## CART   
##   
## 2100 samples  
## 17 predictor  
## 7 classes: 'BRICKFACE', 'CEMENT', 'FOLIAGE', 'GRASS', 'PATH', 'SKY', 'WINDOW'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 1890, 1890, 1890, 1890, 1890, 1890, ...   
## Resampling results across tuning parameters:  
##   
## cp Accuracy Kappa   
## 0.01 0.9104762 0.8955556  
## 0.02 0.9104762 0.8955556  
## 0.03 0.8742857 0.8533333  
## 0.04 0.8709524 0.8494444  
## 0.05 0.8709524 0.8494444  
## 0.06 0.8709524 0.8494444  
## 0.07 0.8709524 0.8494444  
## 0.08 0.8100000 0.7783333  
## 0.09 0.8100000 0.7783333  
## 0.10 0.8100000 0.7783333  
## 0.11 0.8100000 0.7783333  
## 0.12 0.8100000 0.7783333  
## 0.13 0.7876190 0.7522222  
## 0.14 0.7876190 0.7522222  
## 0.15 0.5938095 0.5261111  
## 0.16 0.5676190 0.4955556  
## 0.17 0.1428571 0.0000000  
## 0.18 0.1428571 0.0000000  
## 0.19 0.1428571 0.0000000  
## 0.20 0.1428571 0.0000000  
## 0.21 0.1428571 0.0000000  
## 0.22 0.1428571 0.0000000  
## 0.23 0.1428571 0.0000000  
## 0.24 0.1428571 0.0000000  
## 0.25 0.1428571 0.0000000  
## 0.26 0.1428571 0.0000000  
## 0.27 0.1428571 0.0000000  
## 0.28 0.1428571 0.0000000  
## 0.29 0.1428571 0.0000000  
## 0.30 0.1428571 0.0000000  
## 0.31 0.1428571 0.0000000  
## 0.32 0.1428571 0.0000000  
## 0.33 0.1428571 0.0000000  
## 0.34 0.1428571 0.0000000  
## 0.35 0.1428571 0.0000000  
## 0.36 0.1428571 0.0000000  
## 0.37 0.1428571 0.0000000  
## 0.38 0.1428571 0.0000000  
## 0.39 0.1428571 0.0000000  
## 0.40 0.1428571 0.0000000  
## 0.41 0.1428571 0.0000000  
## 0.42 0.1428571 0.0000000  
## 0.43 0.1428571 0.0000000  
## 0.44 0.1428571 0.0000000  
## 0.45 0.1428571 0.0000000  
## 0.46 0.1428571 0.0000000  
## 0.47 0.1428571 0.0000000  
## 0.48 0.1428571 0.0000000  
## 0.49 0.1428571 0.0000000  
## 0.50 0.1428571 0.0000000  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was cp = 0.02.

Optimal model chosen based on the accuracy with cp=.02.

imageTreeCV = rpart(Class ~ . , method = "class", data = imageSegTranscl, control=rpart.control(cp=.02))  
predictTreeCV= predict(imageTreeCV, newdata=testTranscl, type= "class")  
table(testTranscl$testClass,predictTreeCV)

## predictTreeCV  
## BRICKFACE CEMENT FOLIAGE GRASS PATH SKY WINDOW  
## BRICKFACE 29 0 0 0 0 0 1  
## CEMENT 0 24 1 0 2 0 3  
## FOLIAGE 1 3 15 0 0 0 11  
## GRASS 0 0 0 30 0 0 0  
## PATH 0 0 0 0 30 0 0  
## SKY 0 0 0 0 0 30 0  
## WINDOW 1 1 1 0 0 0 27

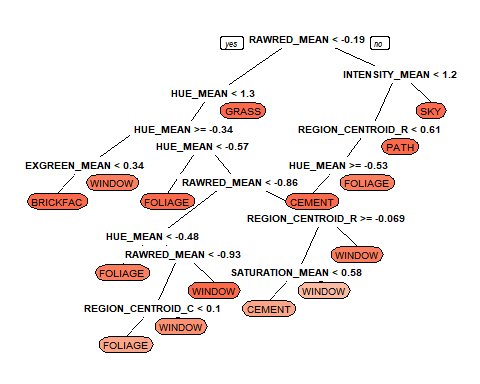
Cross validation does not seem to improve decision tree’s prediction.

### Tree model 3 with repeated cross validation

### Trained decision tree classifier results

## CART   
##   
## 2100 samples  
## 17 predictor  
## 7 classes: 'BRICKFACE', 'CEMENT', 'FOLIAGE', 'GRASS', 'PATH', 'SKY', 'WINDOW'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 3 times)   
## Summary of sample sizes: 1890, 1890, 1890, 1890, 1890, 1890, ...   
## Resampling results across tuning parameters:  
##   
## cp Accuracy Kappa   
## 0.003055556 0.9473016 0.9385185  
## 0.003333333 0.9474603 0.9387037  
## 0.006666667 0.9350794 0.9242593  
## 0.008888889 0.9207937 0.9075926  
## 0.026111111 0.8904762 0.8722222  
## 0.028333333 0.8866667 0.8677778  
## 0.074444444 0.8126984 0.7814815  
## 0.143611111 0.6952381 0.6444444  
## 0.165833333 0.2942857 0.1766667  
## 0.166666667 0.1428571 0.0000000  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was cp = 0.003333333.

### Visualizing the decision tree with prp plot



### Predicting test data

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction BRICKFACE CEMENT FOLIAGE GRASS PATH SKY WINDOW  
## BRICKFACE 29 1 1 0 0 0 1  
## CEMENT 0 28 0 0 1 0 1  
## FOLIAGE 0 1 25 0 0 0 2  
## GRASS 0 0 0 30 0 0 0  
## PATH 0 0 0 0 29 0 0  
## SKY 0 0 0 0 0 30 0  
## WINDOW 1 0 4 0 0 0 26  
##   
## Overall Statistics  
##   
## Accuracy : 0.9381   
## 95% CI : (0.8965, 0.9666)  
## No Information Rate : 0.1429   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9278   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: BRICKFACE Class: CEMENT Class: FOLIAGE  
## Sensitivity 0.9667 0.9333 0.8333  
## Specificity 0.9833 0.9889 0.9833  
## Pos Pred Value 0.9062 0.9333 0.8929  
## Neg Pred Value 0.9944 0.9889 0.9725  
## Prevalence 0.1429 0.1429 0.1429  
## Detection Rate 0.1381 0.1333 0.1190  
## Detection Prevalence 0.1524 0.1429 0.1333  
## Balanced Accuracy 0.9750 0.9611 0.9083  
## Class: GRASS Class: PATH Class: SKY Class: WINDOW  
## Sensitivity 1.0000 0.9667 1.0000 0.8667  
## Specificity 1.0000 1.0000 1.0000 0.9722  
## Pos Pred Value 1.0000 1.0000 1.0000 0.8387  
## Neg Pred Value 1.0000 0.9945 1.0000 0.9777  
## Prevalence 0.1429 0.1429 0.1429 0.1429  
## Detection Rate 0.1429 0.1381 0.1429 0.1238  
## Detection Prevalence 0.1429 0.1381 0.1429 0.1476  
## Balanced Accuracy 1.0000 0.9833 1.0000 0.9194

### Random Forest model 1

##   
## Call:  
## randomForest(formula = Class ~ ., data = imageSeg, nodesize = 25, ntree = 4000)   
## Type of random forest: classification  
## Number of trees: 4000  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 3.1%  
## Confusion matrix:  
## BRICKFACE CEMENT FOLIAGE GRASS PATH SKY WINDOW class.error  
## BRICKFACE 296 1 0 0 0 0 3 0.013333333  
## CEMENT 1 285 1 1 0 0 12 0.050000000  
## FOLIAGE 4 2 284 0 0 1 9 0.053333333  
## GRASS 0 3 0 297 0 0 0 0.010000000  
## PATH 0 2 0 0 298 0 0 0.006666667  
## SKY 0 0 0 0 0 300 0 0.000000000  
## WINDOW 4 5 16 0 0 0 275 0.083333333

### Random Forest model 2

##   
## Call:  
## randomForest(formula = Class ~ ., data = imageSegTranscl, nodesize = 20, ntree = 1000, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 1000  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 3.14%  
## Confusion matrix:  
## BRICKFACE CEMENT FOLIAGE GRASS PATH SKY WINDOW class.error  
## BRICKFACE 296 1 0 0 0 0 3 0.013333333  
## CEMENT 1 284 1 1 0 0 13 0.053333333  
## FOLIAGE 4 2 285 0 0 1 8 0.050000000  
## GRASS 0 2 1 297 0 0 0 0.010000000  
## PATH 0 2 0 0 298 0 0 0.006666667  
## SKY 0 0 0 0 0 300 0 0.000000000  
## WINDOW 4 6 16 0 0 0 274 0.086666667

## BRICKFACE CEMENT FOLIAGE GRASS PATH SKY WINDOW  
## REGION\_CENTROID\_COL 15.43 13.55 32.78 1.27 5.14 1.60 37.84  
## REGION\_CENTROID\_ROW 39.21 89.33 40.99 16.15 156.25 10.94 52.06  
## SHORT\_LINE\_DENSITY\_5 2.83 1.61 0.14 0.00 2.61 0.00 3.23  
## VEDGE\_MEAN 12.77 4.62 16.98 1.85 8.76 2.80 8.63  
## VEDGE\_SD 7.56 2.72 15.79 2.58 5.57 2.91 3.21  
## HEDGE\_MEAN 12.69 8.27 18.15 4.10 10.98 1.34 14.78  
## HEDGE\_SD 6.44 7.67 17.12 3.08 11.63 1.50 13.24  
## INTENSITY\_MEAN 23.90 23.60 25.13 13.44 20.26 16.66 26.13  
## RAWRED\_MEAN 27.72 28.71 34.27 14.98 23.01 15.08 33.03  
## RAWBLUE\_MEAN 21.30 18.57 21.18 12.58 18.15 16.86 23.05  
## RAWGREEN\_MEAN 23.38 23.10 19.55 12.77 18.78 14.15 22.34  
## EXRED\_MEAN 32.91 12.98 25.71 6.09 12.06 3.08 28.36  
## EXBLUE\_MEAN 18.15 13.14 18.21 14.11 15.61 5.32 21.03  
## EXGREEN\_MEAN 49.06 26.38 28.68 25.08 20.36 11.14 36.94  
## VALUE\_MEAN 21.70 19.63 21.28 12.59 16.26 17.97 24.12  
## SATURATION\_MEAN 21.49 22.05 33.38 7.42 16.34 5.50 15.23  
## HUE\_MEAN 50.87 39.25 79.07 24.86 21.46 11.64 60.12  
## MeanDecreaseAccuracy MeanDecreaseGini  
## REGION\_CENTROID\_COL 38.30 34.77  
## REGION\_CENTROID\_ROW 84.44 255.50  
## SHORT\_LINE\_DENSITY\_5 4.82 0.18  
## VEDGE\_MEAN 21.35 17.59  
## VEDGE\_SD 15.55 9.76  
## HEDGE\_MEAN 24.06 37.17  
## HEDGE\_SD 19.83 25.66  
## INTENSITY\_MEAN 30.03 135.51  
## RAWRED\_MEAN 37.72 173.19  
## RAWBLUE\_MEAN 26.50 111.22  
## RAWGREEN\_MEAN 28.66 104.24  
## EXRED\_MEAN 41.94 71.56  
## EXBLUE\_MEAN 26.64 68.25  
## EXGREEN\_MEAN 47.87 174.09  
## VALUE\_MEAN 26.51 112.36  
## SATURATION\_MEAN 35.35 79.50  
## HUE\_MEAN 65.20 279.11

We can see the oob estimate of error rate is increasing and decreasing..

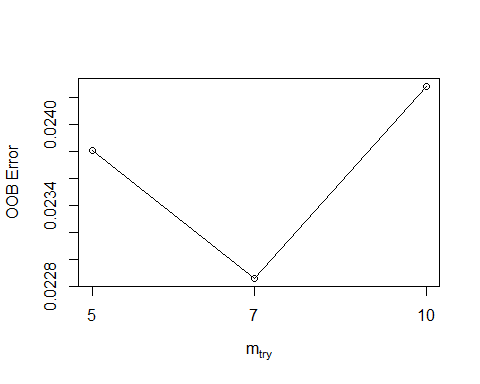
### Random Forest on test data

##   
## Call:  
## randomForest(formula = testClass ~ ., data = testTranscl, ntree = 500)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 5.24%  
## Confusion matrix:  
## BRICKFACE CEMENT FOLIAGE GRASS PATH SKY WINDOW class.error  
## BRICKFACE 28 0 0 0 0 0 2 0.06666667  
## CEMENT 1 26 0 0 0 0 3 0.13333333  
## FOLIAGE 1 0 29 0 0 0 0 0.03333333  
## GRASS 0 0 0 30 0 0 0 0.00000000  
## PATH 0 0 0 0 30 0 0 0.00000000  
## SKY 0 0 0 0 0 30 0 0.00000000  
## WINDOW 1 1 2 0 0 0 26 0.13333333

On test data too, the prediction is not convincing.

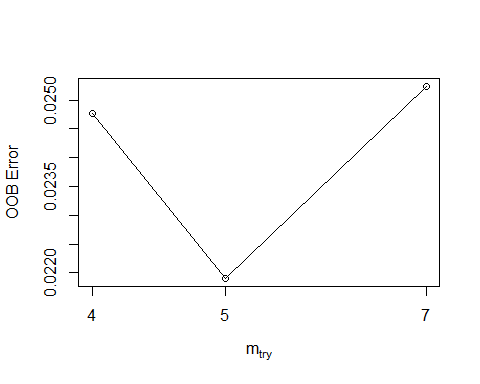
## RandomForest with cross validation

## mtry = 7 OOB error = 2.29%   
## Searching left ...  
## mtry = 5 OOB error = 2.38%   
## -0.04166667 1e-05   
## Searching right ...  
## mtry = 10 OOB error = 2.43%   
## -0.0625 1e-05



## mtry OOBError  
## 5.OOB 5 0.02380952  
## 7.OOB 7 0.02285714  
## 10.OOB 10 0.02428571

## mtry = 5 OOB error = 2.19%   
## Searching left ...  
## mtry = 4 OOB error = 2.48%   
## -0.1304348 1e-05   
## Searching right ...  
## mtry = 7 OOB error = 2.52%   
## -0.1521739 1e-05



## mtry OOBError  
## 4.OOB 4 0.02476190  
## 5.OOB 5 0.02190476  
## 7.OOB 7 0.02523810

Picking the ntree and mtry from the results of tuneRF function:

##   
## Call:  
## randomForest(formula = Class ~ ., data = imageSegTranscl, mtry = 5, ntree = 400)   
## Type of random forest: classification  
## Number of trees: 400  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 2.38%  
## Confusion matrix:  
## BRICKFACE CEMENT FOLIAGE GRASS PATH SKY WINDOW class.error  
## BRICKFACE 298 0 0 0 0 0 2 0.006666667  
## CEMENT 1 290 1 1 1 0 6 0.033333333  
## FOLIAGE 4 3 284 0 0 1 8 0.053333333  
## GRASS 0 2 0 298 0 0 0 0.006666667  
## PATH 0 0 0 0 300 0 0 0.000000000  
## SKY 0 0 0 0 0 300 0 0.000000000  
## WINDOW 3 2 15 0 0 0 280 0.066666667

### Predicting the testdata using this model

## predictForestCV  
## BRICKFACE CEMENT FOLIAGE GRASS PATH SKY WINDOW  
## BRICKFACE 30 0 0 0 0 0 0  
## CEMENT 0 30 0 0 0 0 0  
## FOLIAGE 0 0 30 0 0 0 0  
## GRASS 0 0 0 30 0 0 0  
## PATH 0 0 0 0 30 0 0  
## SKY 0 0 0 0 0 30 0  
## WINDOW 0 0 1 0 0 0 29

## Conclusion

|  |  |  |
| --- | --- | --- |
| **Technique/Model** | **Parameters** | **Accuracy on test set** |
| SVM | Cost =100  Gamma =0.5 | 99.5% |
| Random Forest | node=25  ntree=1000 | 97% |
| Random Forest with cross validation | mtry=5  ntree=400 | 98% |
| glm | prediction>.95 | 98.5% |
| Decision Tree | Using “rpart”  Minbucket = 25 | 88% |
| Decision Tree with Cross Validation | Using “rpart”  1. Method = cv  Number of folds=10  .cp=0.01:0.5  2. method=repeatedcv  number=10  repeats=3  split=gini/information  tuneLength=10 | 97%  93.8% |

Since this dataset is multiple classification type and the features are not linearly correlated, **SVM model** with the parameters *cost=10* and *gamma=0.5* and the **Random Fores**t algorithm with *cross validation* works the best.