HW4

Han Ambrose

May 1, 2021

Question 2

Question 2a

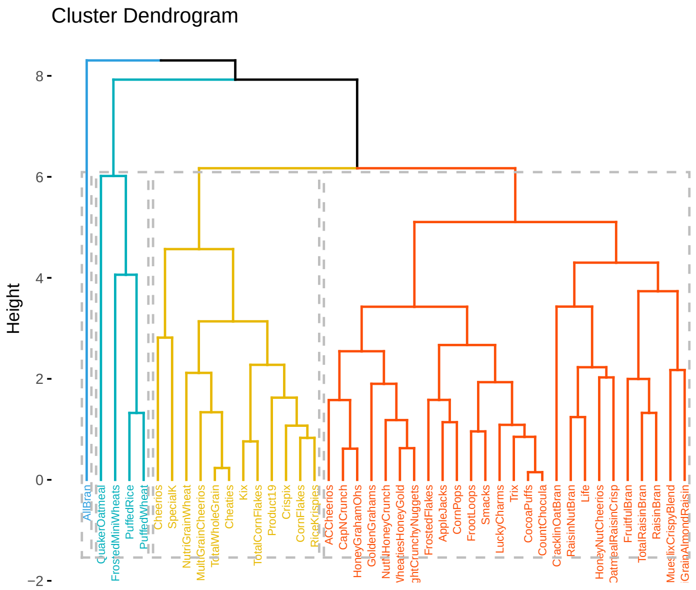


Figure 1: Complete Link

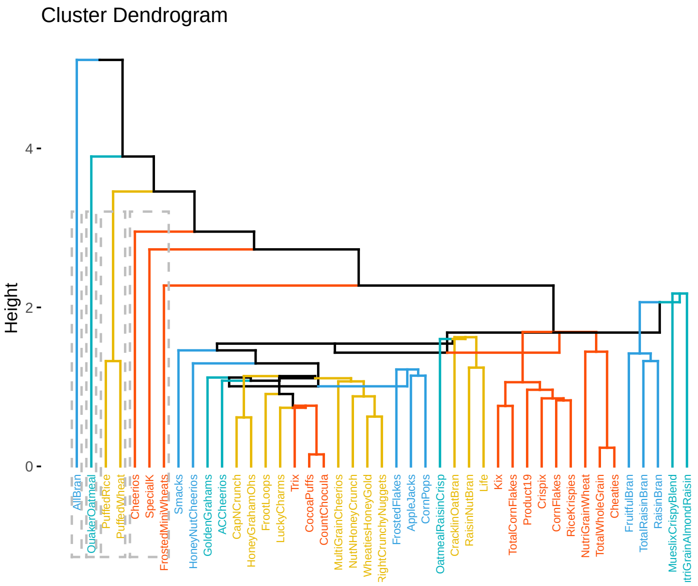


Figure 2: Centroid Link

Question 2b

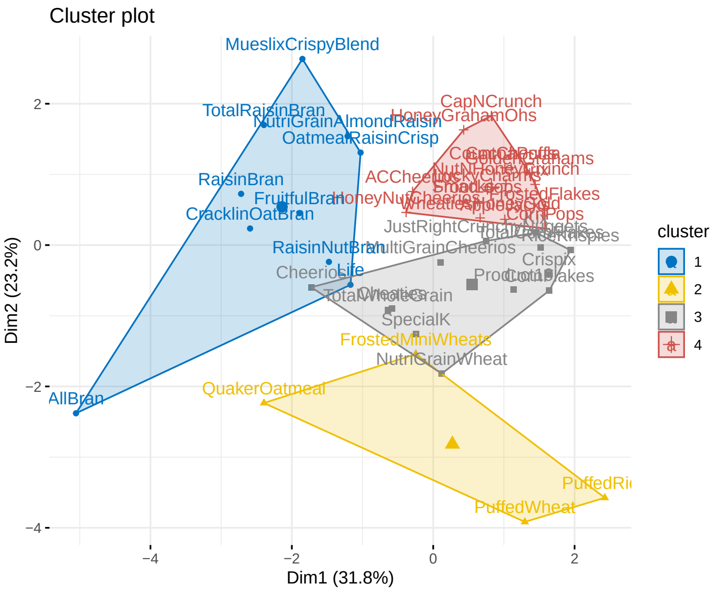


Figure 3: Kmeans with K =4

Question 2c

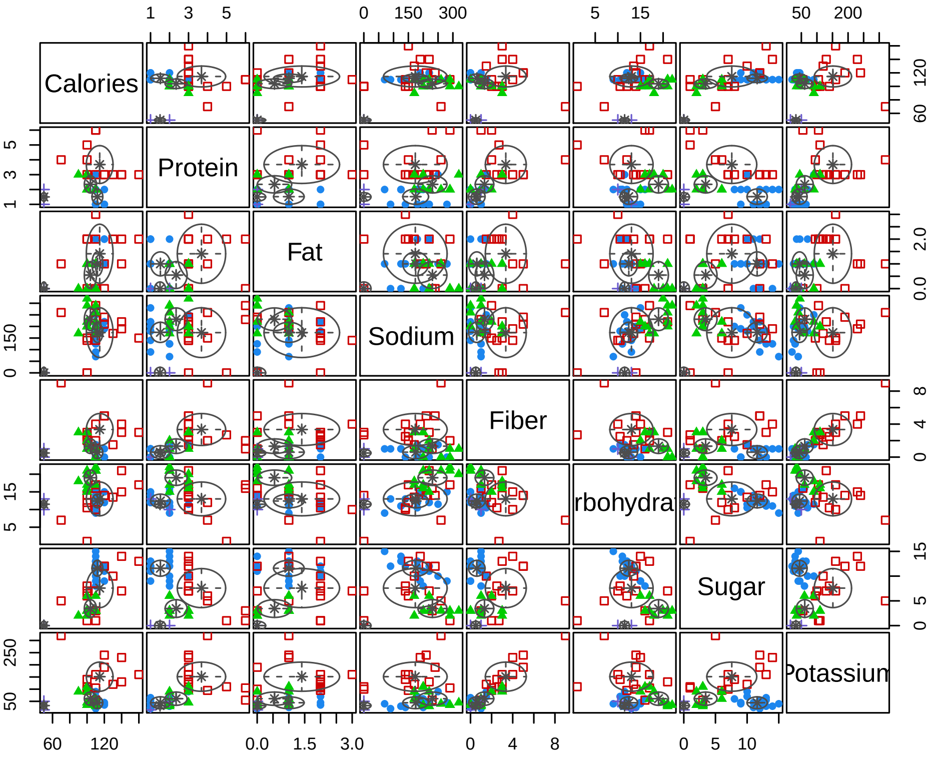


Figure 4: Classification With 4 clusters (Model Base with K =4)

Question 2d

We can see that there are clusters of healthy cereals/adults cereals versus kids cereals. There is also a cluster of wheat and oatmeals cereal. All bran/nut brans cereals seem to be different then the rest.

Since complete link method is space dilating, we see that clusters are more far apart. However, centroid and k-means seems to be more evenly spread out or space conserving.

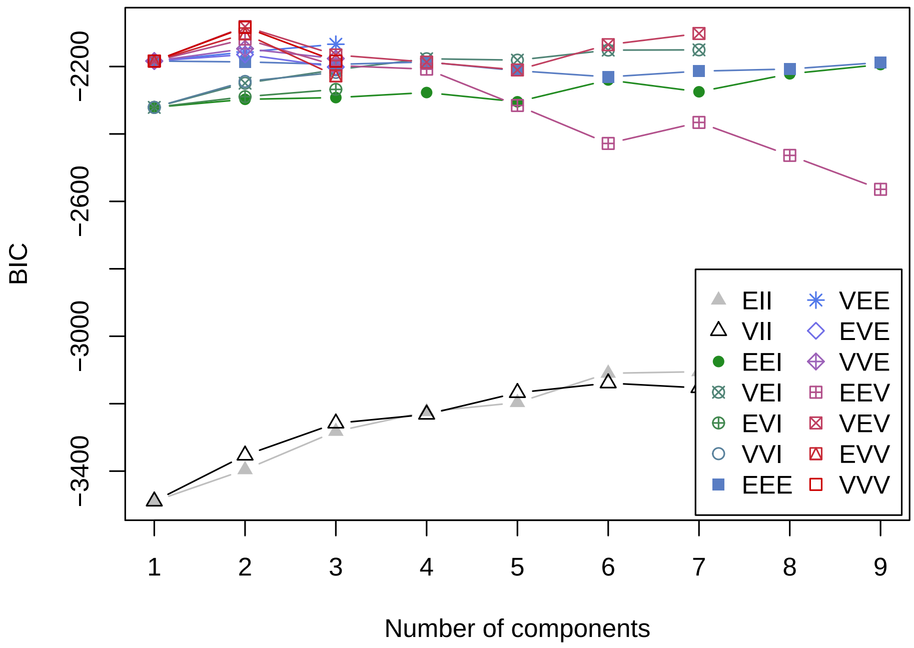
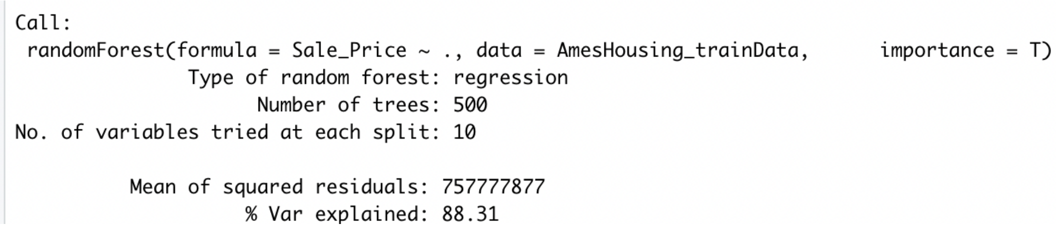


Figure 5: Model Based using BIC

Question 3

Question 3 - a,b,c,d


```
Call:
 randomForest(formula = Sale_Price ~ ., data = AmesHousing_trainData,      importance = T)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 10

          Mean of squared residuals: 757777877
                    % Var explained: 88.31
```

Figure 6: Random Forest Output

Question 3 - e

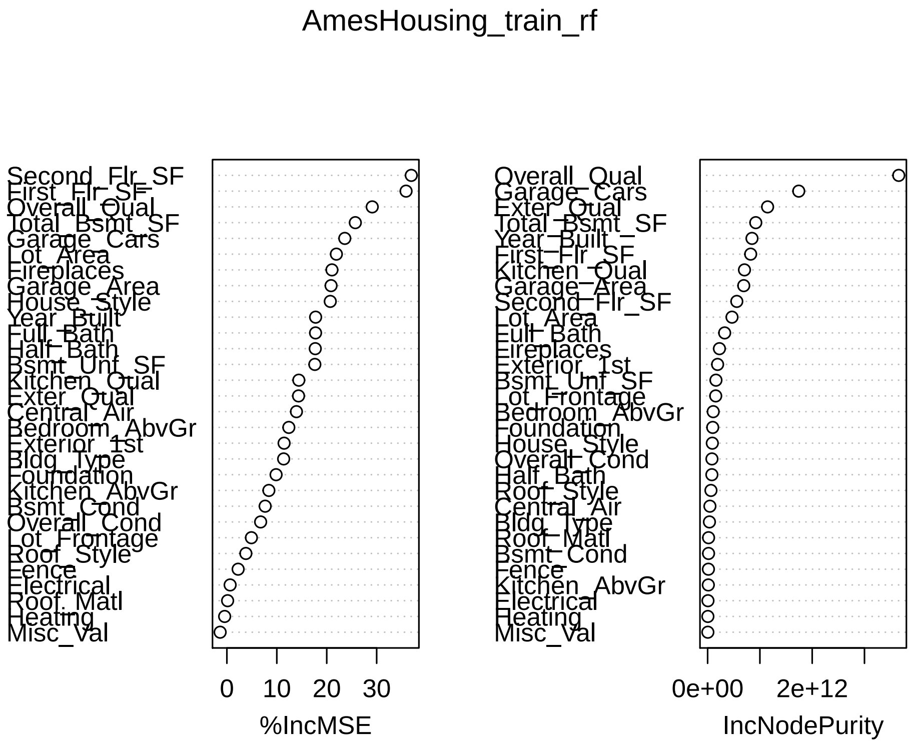


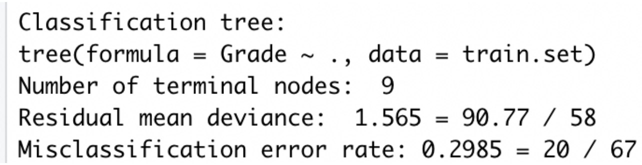
Figure 7: Importance

Question 3 - f

Mean Square Error on the testing set is 632087196 by using formula below
mean((AmesHousing_testData\$Sale_Price - housing_pred)^{2})

Question 4

Question 4 - a,b



```
Classification tree:
tree(formula = Grade ~ ., data = train.set)
Number of terminal nodes:  9
Residual mean deviance:  1.565 = 90.77 / 58
Misclassification error rate: 0.2985 = 20 / 67
```

Figure 8: Summary of tree model

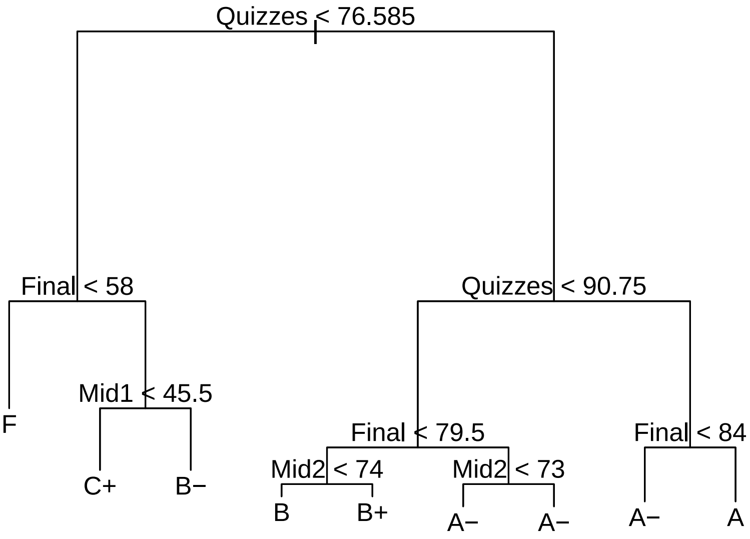
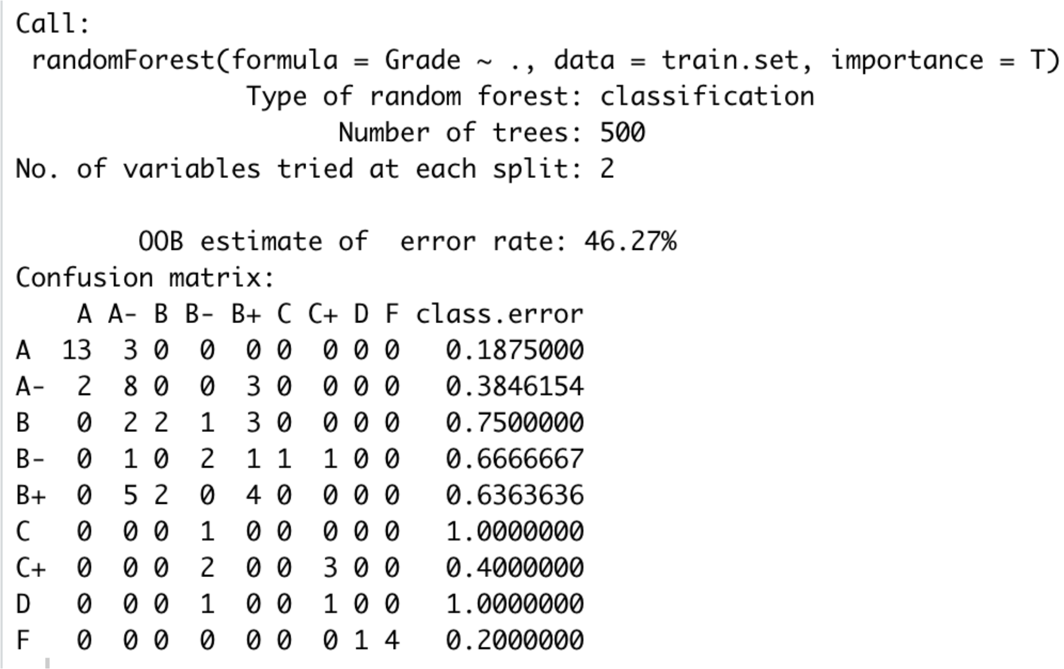


Figure 9: Plot of tree

Quizzes that is less than or greater than 76.5 is the most important score as it is the first split

Question 4 - c,d



```
Call:
 randomForest(formula = Grade ~ ., data = train.set, importance = T)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 46.27%
Confusion matrix:
    A A- B B- B+ C C+ D F class.error
A  13  3 0  0  0 0  0 0 0   0.1875000
A-  2  8 0  0  3 0  0 0 0   0.3846154
B   0  2 2  1  3 0  0 0 0   0.7500000
B-  0  1 0  2  1 1  1 0 0   0.6666667
B+  0  5 2  0  4 0  0 0 0   0.6363636
C   0  0 0  1  0 0  0 0 0   1.0000000
C+  0  0 0  2  0 0  3 0 0   0.4000000
D   0  0 0  1  0 0  1 0 0   1.0000000
F   0  0 0  0  0 0  0 1 4   0.2000000
```

Figure 10: Random Forest model

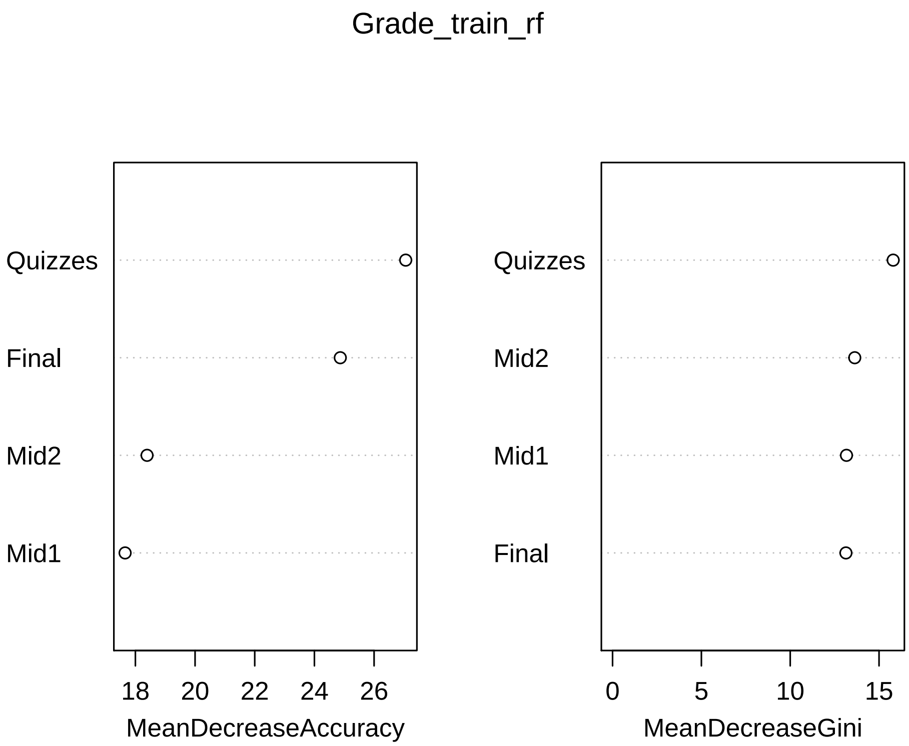
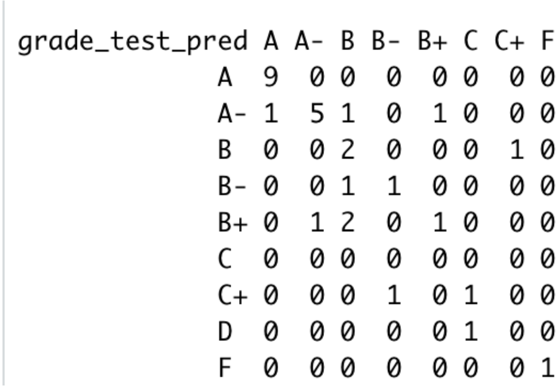


Figure 11: Importance

Quizzes is the most important feature

Question 4 - e



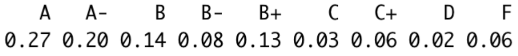
```
grade_test_pred A A- B B- B+ C C+ F
             A  9  0 0  0  0 0  0 0
             A- 1  5 1  0  1 0  0 0
             B  0  0 2  0  0 0  1 0
             B- 0  0 1  1  0 0  0 0
             B+ 0  1 2  0  1 0  0 0
             C  0  0 0  0  0 0  0 0
             C+ 0  0 0  1  0 1  0 0
             D  0  0 0  0  0 1  0 0
             F  0  0 0  0  0 0  0 1
```

Figure 12: Testing accuracy

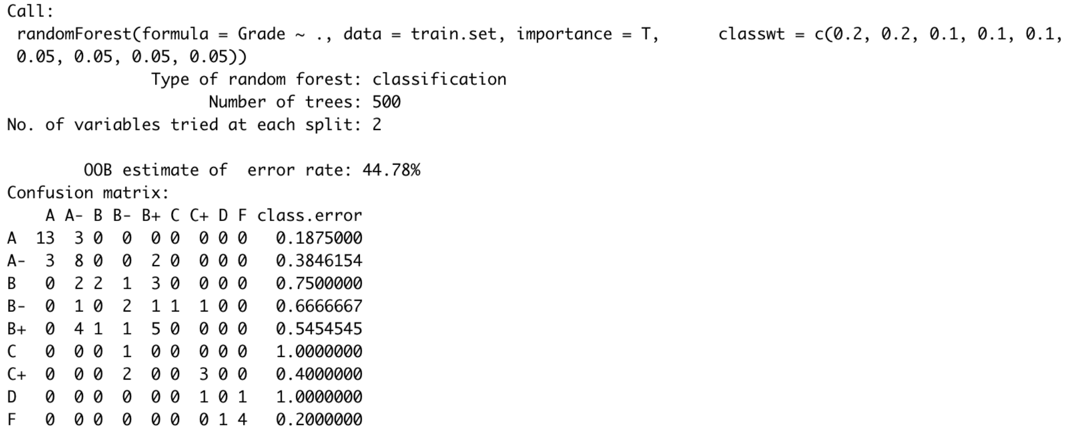
accuracy_m1 = mean(grade_test_pred == test.set$Grade)
The overall accuracy is 63%

Question 4 - f

Below is the proportions of class from the dataset given. Grade A is overrepresented.



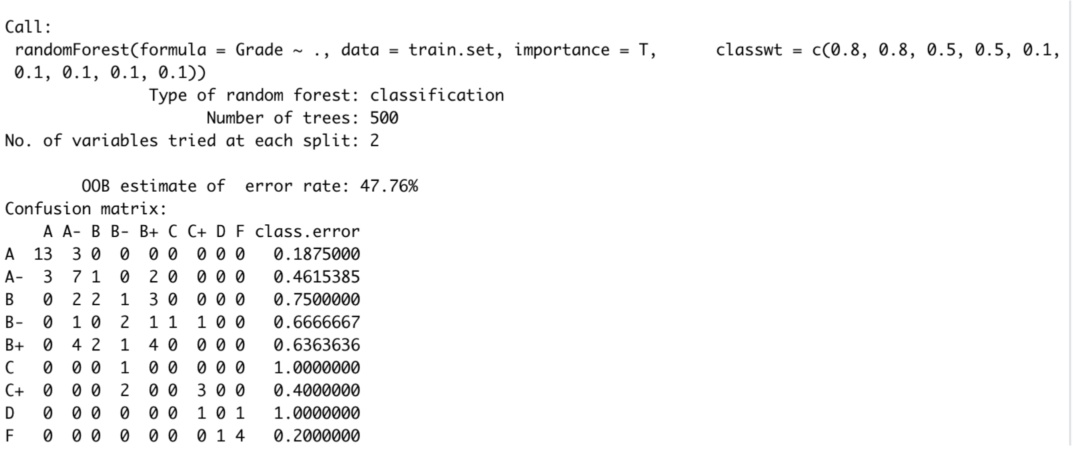
```
   A   A-    B   B-   B+    C   C+    D    F 
0.27 0.20 0.14 0.08 0.13 0.03 0.06 0.02 0.06 
```



```
Call:
 randomForest(formula = Grade ~ ., data = train.set, importance = T,      classwt = c(0.2, 0.2, 0.1, 0.1, 0.1, 
 0.05, 0.05, 0.05, 0.05)) 
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 44.78%
Confusion matrix:
    A A- B B- B+ C C+ D F class.error
A  13  3 0  0  0 0  0 0 0   0.1875000
A-  3  8 0  0  2 0  0 0 0   0.3846154
B   0  2 2  1  3 0  0 0 0   0.7500000
B-  0  1 0  2  1 1  1 0 0   0.6666667
B+  0  4 1  1  5 0  0 0 0   0.5454545
C   0  0 0  1  0 0  0 0 0   1.0000000
C+  0  0 0  2  0 0  3 0 0   0.4000000
D   0  0 0  0  0 0  1 0 1   1.0000000
F   0  0 0  0  0 0  0 1 4   0.2000000
```

Figure 13: using classwt for imbalance class



```
Call:
 randomForest(formula = Grade ~ ., data = train.set, importance = T,      classwt = c(0.8, 0.8, 0.5, 0.5, 0.1, 
 0.1, 0.1, 0.1, 0.1)) 
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 47.76%
Confusion matrix:
    A A- B B- B+ C C+ D F class.error
A  13  3 0  0  0 0  0 0 0   0.1875000
A-  3  7 1  0  2 0  0 0 0   0.4615385
B   0  2 2  1  3 0  0 0 0   0.7500000
B-  0  1 0  2  1 1  1 0 0   0.6666667
B+  0  4 2  1  4 0  0 0 0   0.6363636
C   0  0 0  1  0 0  0 0 0   1.0000000
C+  0  0 0  2  0 0  3 0 0   0.4000000
D   0  0 0  0  0 0  1 0 1   1.0000000
F   0  0 0  0  0 0  0 1 4   0.2000000
```

Figure 14: using classwt for imbalance class

I tried different weights and notice that the testing accuracy improve better when grade A and B were assigned a lot heavier weight

Once applied to testing data, the testing accuracy score improved and increased slightly to 67%