Bryson Cook
ISYE6501, Spring 2018
HW7

Question 10.1
Using the same crime data set as in Questions 8.2 and 9.1, find the best model you can using
(a) a regression tree model, and
(b) a random forest model.
In R, you can use the `tree` package or the `rpart` package, and the `randomForest` package. For each model, describe one or two qualitative takeaways you get from analyzing the results (i.e., don't just stop when you have a good model, but interpret it too).

(a). Using the rpart function and splitting the original data into training and test sets, I created a regression tree as shown in Figure 1. The tree had three leaves, with the decisions based on the Po1 and Po2 factors. However, when using the predict function on the test data the quality of the model is very poor, with 1-SSE/SST actually being negative.

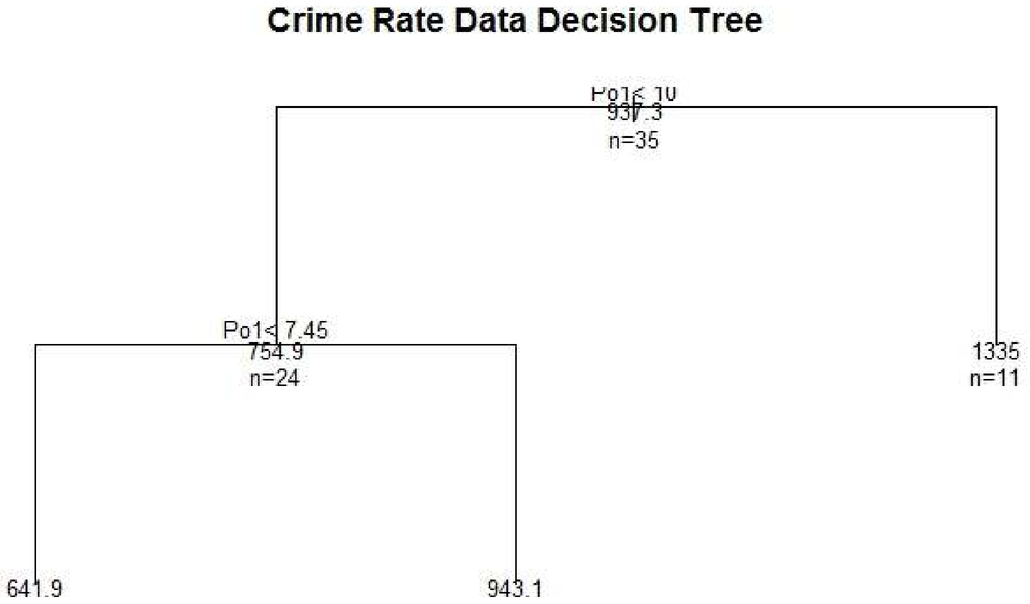


Figure 1. Original Crime Rate Regression Tree

I then pruned the tree using the lowest cross-validation error from the original tree's summary, and the tree was reduces to two leaves, which the decision still based on the Po1 factor. This tree is shown in Figure 2.

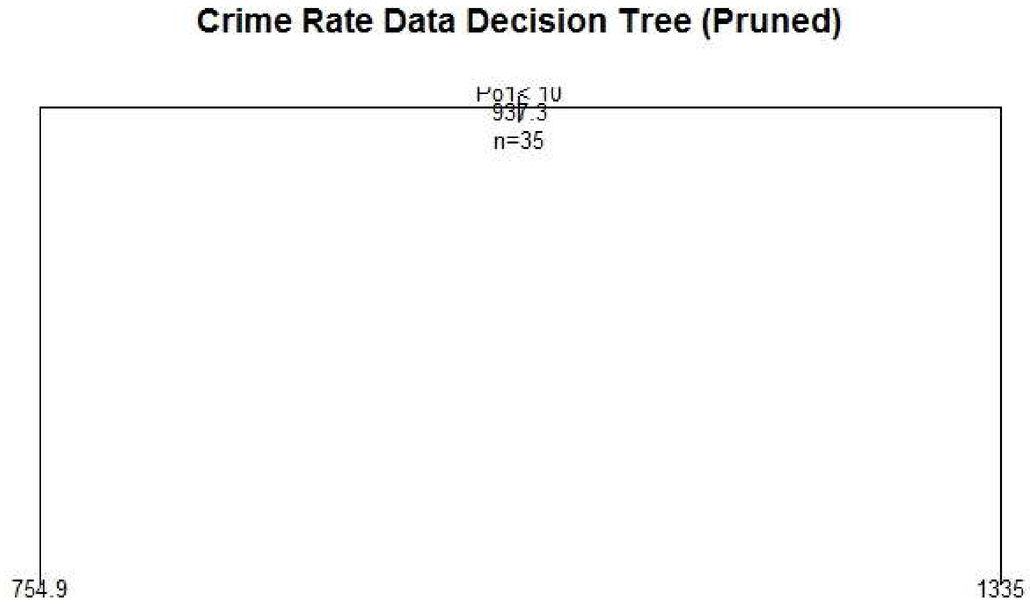


Figure 2. Pruned Crime Rate Regression Tree

Unfortunately this model is even worse than the original. I then created regression models for each leaf and reduced the number of factors based on p-values. In terms of leaf 2, only two factors (Ineq and Wealth) had a p-value under .1, so a limit of 0.4 was used. For leaf 3, the model is so over fit that 5 of the coefficients were not defined, so I eliminated those. The resultant adjusted R2 for leaves were 0.2919 for leaf 2 and .4952. My two takeaways from the regression tree model are:

1. I was surprised to see that the Po1 variable chosen by the function as the decision factor for the tree. While Po1 was shown by past homeworks and data preparation tools to be an important factor, nothing indicated that it had such a large impact on explaining the variance.
2. The original regression tree model is very poor, most likely due to over fitting. By pruning the tree and removing factors from the resultant leaves' regression models, the model quality can be increased. However, it seems that regression tree models are simply not meant for such small amounts of data, especially when there so many factors relative to the amount of data (15 factors vs 47 data points) as overfitting become a very large issue.

(b) Splitting the data into training and testing sets, I used the RandomForest package to create a random forest for the crime data. Evaluating the model against the training data, we get 1- sse/sst = 0.425. However, when testing the model on the test set, we get 1- sse/sst = 0.204, which, as expected, is lower than the R2 of the training data, but is still much higher than the single regression tree. My takeaways from the RandomForest model are:

1. As expected, the random forest was able to limit the effect of overfitting by introducing randomness. Even though the model probably should not be considered "good", it is still much better quality than the single regression tree.

2. When looking at the importance of the factors in the random forest, shown in Figure 3, it is interesting to see the Po1 and Po2 variables at the top. As noted earlier, while Po1 was shown by past homeworks and data preparation tools to be an important factor, nothing indicated that it had such a large impact on explaining the variance. We noted from the PCA analysis that Po1 and Po2 were highly correlated, so see them together is not a surprise.

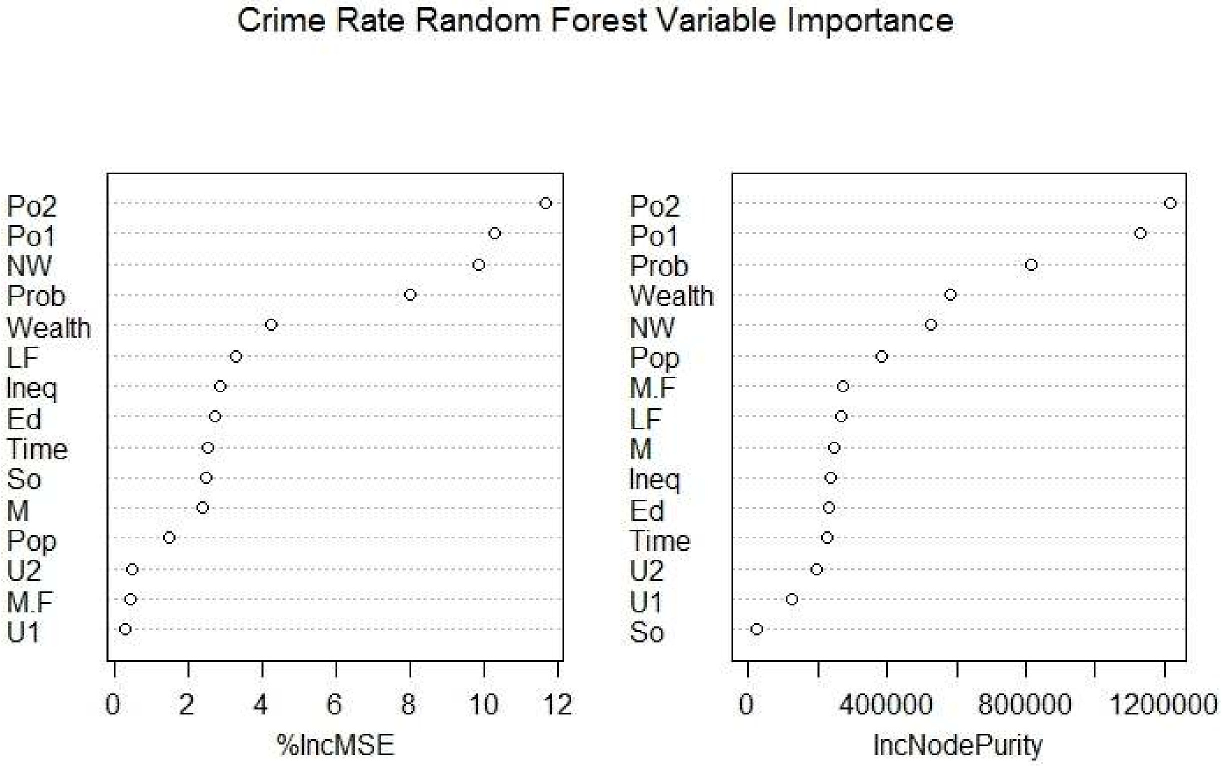


Figure 3. Factor Importance in Random Forest Model

Question 10.2
Describe a situation or problem from your job, everyday life, current events, etc., for which a logistic regression model would be appropriate. List some (up to 5) predictors that you might use.

With all of the news about gerrymandering, I would think that logistic regression was/is used by politicians or researchers to classify political affiliation, or at least the probability of political leanings, which is then in turn used to create the new congressional districts (I'm not judging the legitimacy of the current district maps, just noting that logistic regression might have been used). I would assume they would take into account average household income, median home prices, average age in community/neighborhood, and party affiliation of certain local representatives (city, county, and state legislative representatives) would all be likely predictors to how the people in that area may vote.

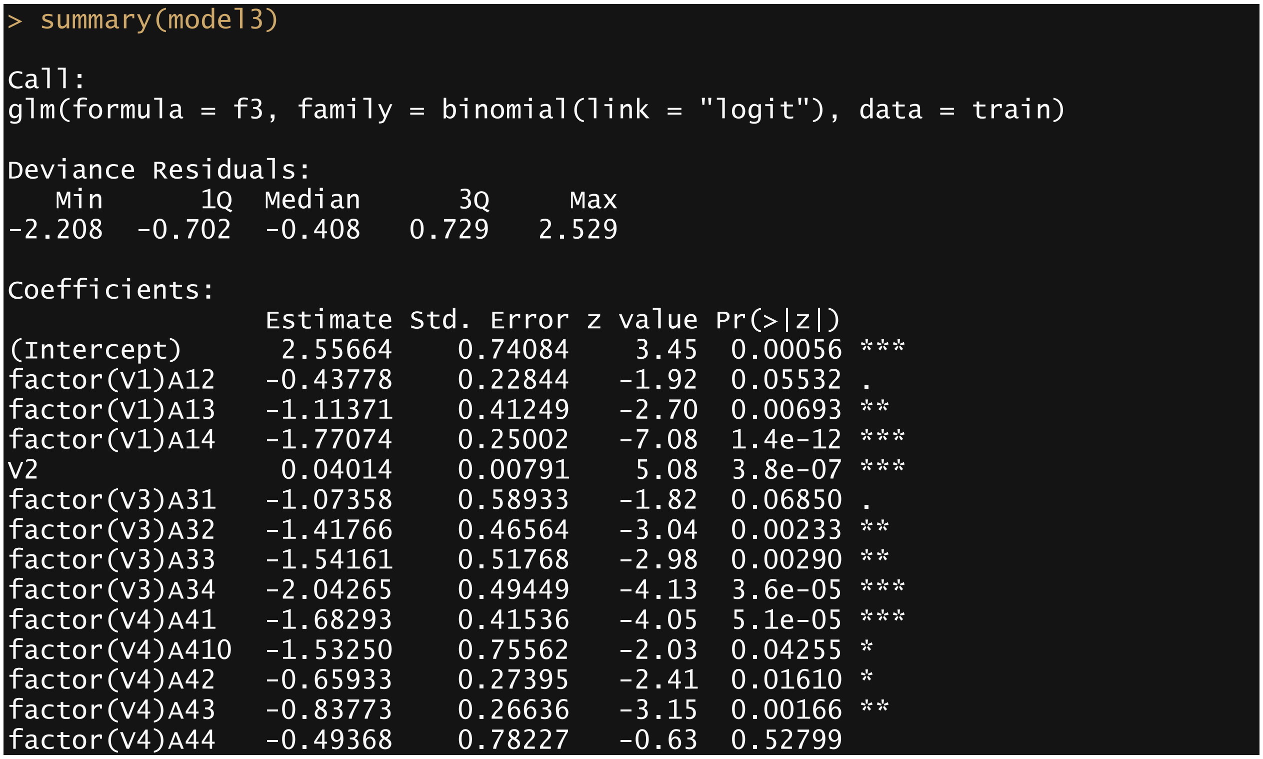
Question 10.3

1. **Using the GermanCredit data set use logistic regression to find a good predictive model for whether credit applicants are good credit risks or not. Show your model (factors used and their coefficients), the software output, and the quality of fit. You can use the `glm` function in R. To get a logistic regression (logit) model on data where the response is either zero or one, use `family=binomial(link=”logit”)` in your `glm` function call.**
2. **Because the model gives a result between 0 and 1, it requires setting a threshold probability to separate between “good” and “bad” answers. In this data set, they estimate that incorrectly identifying a bad customer as good, is 5 times worse than incorrectly classifying a good customer as bad. Determine a good threshold probability based on your model.**

1. The data was split into training and test sets and the glm() function was used to create logistic regression model of the data. When reviewing the summary of the original model, many of the factor have high p-values and do not seem relevant. Therefore, all factors with p-values >.05 were removed and a new model was created. This was repeated, and the formula for the final model is:

Y = 2.5566 + -0.4378*(V1)A12 + -1.1137*(V1)A13 + -1.7707*(V1)A14 + 0.0401*V2 + -1.0736*(V3)A31 + -1.4177*(V3)A32 + -1.5416*(V3)A33 + -2.0427*(V3)A34 + -1.6829*(V4)A41 + -1.5325*(V4)A410 + -0.6593*(V4)A42 + -0.8377*(V4)A43 + -0.4937*(V4)A44 + -0.537*(V4)A45 + 0.3954*(V4)A46 + -2.0052*(V4)A48 + -0.6982*(V4)A49 + 0.1863*V8 + -0.4152*(V9)A92 + -0.918*(V9)A93 + -0.5654*(V9)A94 + 0.4465*(V10)A102 + -1.1753*(V10)A103 + -0.1864*(V14)A142

A summary of the model is shown below:

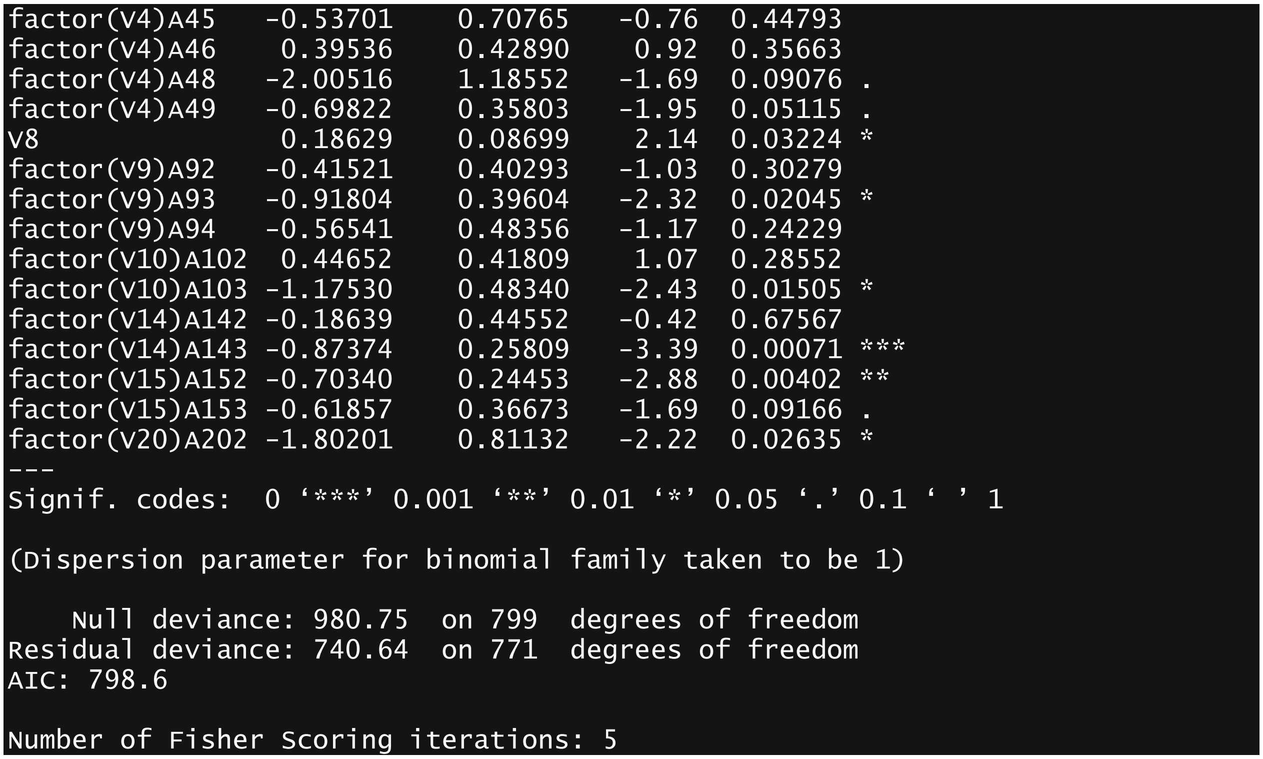


```
> summary(model3)

Call:
glm(formula = f3, family = binomial(link = "logit"), data = train)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.208  -0.702  -0.408   0.729   2.529

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      2.55664    0.74084    3.45  0.00056 ***
factor(V1)A12   -0.43778    0.22844   -1.92  0.05532 .
factor(V1)A13   -1.11371    0.41249   -2.70  0.00693 **
factor(V1)A14   -1.77074    0.25002   -7.08  1.4e-12 ***
V2               0.04014    0.00791    5.08  3.8e-07 ***
factor(V3)A31   -1.07358    0.58933   -1.82  0.06850 .
factor(V3)A32   -1.41766    0.46564   -3.04  0.00233 **
factor(V3)A33   -1.54161    0.51768   -2.98  0.00290 **
factor(V3)A34   -2.04265    0.49449   -4.13  3.6e-05 ***
factor(V4)A41   -1.68293    0.41536   -4.05  5.1e-05 ***
factor(V4)A410  -1.53250    0.75562   -2.03  0.04255 *
factor(V4)A42   -0.65933    0.27395   -2.41  0.01610 *
factor(V4)A43   -0.83773    0.26636   -3.15  0.00166 **
factor(V4)A44   -0.49368    0.78227   -0.63  0.52799
```



```
factor(V4)A45   -0.53701    0.70765   -0.76  0.44793
factor(V4)A46    0.39536    0.42890    0.92  0.35663
factor(V4)A48   -2.00516    1.18552   -1.69  0.09076 .
factor(V4)A49   -0.69822    0.35803   -1.95  0.05115 .
V8               0.18629    0.08699    2.14  0.03224 *
factor(V9)A92   -0.41521    0.40293   -1.03  0.30279
factor(V9)A93   -0.91804    0.39604   -2.32  0.02045 *
factor(V9)A94   -0.56541    0.48356   -1.17  0.24229
factor(V10)A102  0.44652    0.41809    1.07  0.28552
factor(V10)A103 -1.17530    0.48340   -2.43  0.01505 *
factor(V14)A142 -0.18639    0.44552   -0.42  0.67567
factor(V14)A143 -0.87374    0.25809   -3.39  0.00071 ***
factor(V15)A152 -0.70340    0.24453   -2.88  0.00402 **
factor(V15)A153 -0.61857    0.36673   -1.69  0.09166 .
factor(V20)A202 -1.80201    0.81132   -2.22  0.02635 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 980.75  on 799  degrees of freedom
Residual deviance: 740.64  on 771  degrees of freedom
AIC: 798.6

Number of Fisher Scoring iterations: 5
```

Using the cv.glm function we see that model3 has a cross-validation estimate of prediction error of 0.168 and from above, we see an AIC of 798.6.

2. Using the test data and the predict.glm function, we can now pick the best threshold. Using a for loop to loop through threshold values between 0.01 and 0.99 by .01 steps, the total cost at each threshold was calculated noting that Total Cost = FP*5 + FN*1. The FP and FN values were obtained by the confusionMatix() function. From this we see the minimum total cost of 57 is found when a threshold of 0.86 is used. The ROC AUC for this threshold is on 0.509, which is bad. However, since we aren’t trying to come up with the “best prediction” but rather “the lowest total cost”, this is acceptable.