James Baker

12/7/17

CMSC478-01

[bjames6@umbc.edu](mailto:bjames6@umbc.edu)

**Challenge Problem 2 Model Description**

Using random forests, I made a prediction model for the given text training data. I start by randomly selecting a relatively small number (65) of observations and put it in a smaller training set. Then I make every column a factor since they are all categorical variables and not quantitative; even the response variable class. Since random forests cannot take factors with more than 53 levels of predictors, (which is why I chose a small random sample of the data), I sift through the smaller training set to remove any predictors with more than 53 levels in their factors. I chose a small training set, so it would more likely than not, not take out any predictors. Then I ran random forest and messed with the random seed enough to get an error rate of 26.15%.

This error rate was extremely lucky. Larger training sets gave factors with too large of levels, so it excluded predictors. This was leading to very poor error rates because if the rando sample size was too high, it would exclude more and more predictors, because those predictors would have too many levels for the randomForest function. I really did not want to exclude predictors since they were representative of a sequence, so I instead made the tradeoff to reduce the number of observations from 20000, to a random 65 (which still bothers me). I feel like if I had a few more days, I would try the cut function, to get a representation of all or a lot of the data, but have a suitable amount of levels for the random forest function. In short, though I am not positive, I feel I could do a better model than what I am turning in now. It will probably be updated when I turn in my predictions.

lm.fitx = lm(x ~ close1\*cLabel1+close2\*cLabel2+close3\*cLabel3, data=myPaths.reg)

lm.fity = lm(y ~ close1\*cLabel1+close2\*cLabel2+close3\*cLabel3, data=myPaths.reg)

**74 coefficients, Adj. R2 was .93 for y and .93 for x**

lm.fitx = lm(x ~ close1++close2+close3+close1:cLabel1+close2:cLabel2+close3:cLabel3, data=myPaths.reg)

lm.fity = lm(y ~ close1++close2+close3+close1:cLabel1+close2:cLabel2+close3:cLabel3, data=myPaths.reg)

**39 coefficients, Adj. R2 was .88 for x and .87 for y**

lm.fitx = lm(x ~ close1+close2+close3+cLabel1+cLabel2+cLabel3, data=myPaths.reg)

lm.fity = lm(y ~ close1+close2+close3+cLabel1+cLabel2+cLabel3, data=myPaths.reg)

**39 coefficients, Adj. R2 was .92 for x and .89 for y**

lm.fitx = lm(x ~ close1\*cLabel1+close2\*cLabel2, data=myPaths.reg)

lm.fity = lm(y ~ close1\*cLabel1+close2\*cLabel2, data=myPaths.reg)

**51 coefficients, Adj. R2 was .90 for x and .90 for y**

lm.fitx = lm(x ~ close1+cLabel1+close2+cLabel2, data=myPaths.reg)

lm.fity = lm(y ~ close1+cLabel1+close2+cLabel2, data=myPaths.reg)

**27 coefficients, Adj. R2 was .87 for x and .89 for y**

lm.fitx = lm(x ~ close1\*cLabel1, data=myPaths.reg)

lm.fity = lm(y ~ close1\*cLabel1, data=myPaths.reg)

**26 coefficients, Adj. R2 was .81 for x and .83 for y**

lm.fitx = lm(x ~ close1+cLabel1, data=myPaths.reg)

lm.fity = lm(y ~ close1+cLabel1, data=myPaths.reg)

**14 coefficients, Adj. R2 was .76 for x and .8 for y**