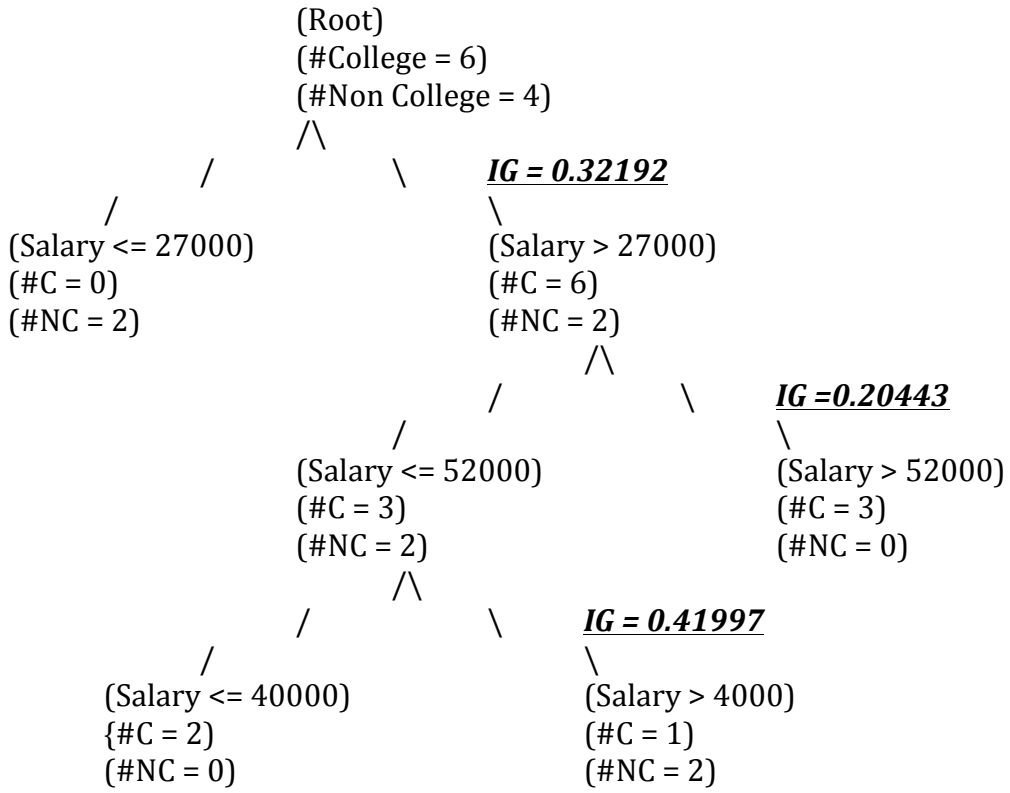
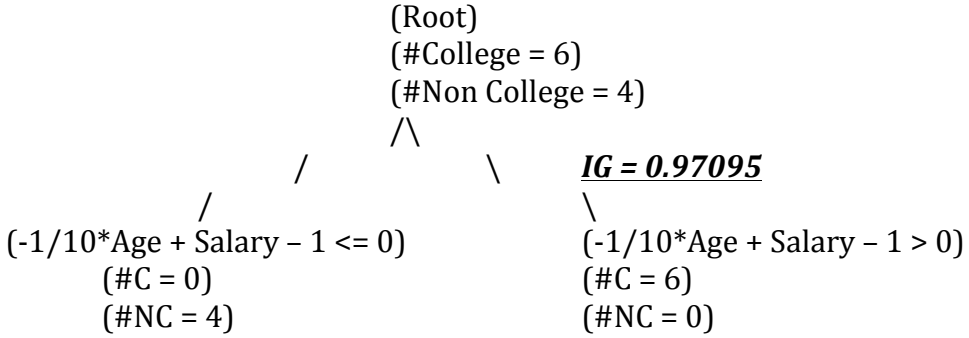
Yuechen Zhao
1126125
CSE 446 HW1

1. Decision Trees

1) Since this there are very few inputs, I decided to prune the tree by limiting depth of the tree. In this case, I allowed a depth of 3. My code initializes the upper and lower bounds of age and salary to largest and smallest values possible, and as the tree grows, the programmer manually updates the upper and lower bounds.



2) Alpha = -1/10, and Beta = 1.



3) An advantage of multivariate decision trees is that once the alpha and beta are determined, the code is very easy to write, and the code does not contain a lot of iterations. However, the disadvantage of this method, which is also its meat, is that programmers need to observe the scatterplot, and try drawing a few lines to split the points. Sometimes it could be very hard to draw a splitting line just by observing, so several alpha and beta values should be tried using code, which could have an infinite number of possible combinations.

2. MLE

1) Give the log-likelihood function of G given λ
   Likelihood(G| λ) = $\Pi_{i = 1 \text{ to } n} \lambda^{Gi}$ / Gi ! * e^(- λ) = λ^(ΣGi) / ΠGi ! * e^(- λn)
   Log-Likelihood = ΣGi * log λ – λn – log(ΠGi !)

2) dLL/d λ = ΣGi / λ – n
   Set dLL/d λ = 0
   MLE(λ) = ΣGi / n

3) MLE(λ) = (6 + 4 + 2 + 7 + 5 + 1 + 2 + 5) / 8 = 4

3. Regularization Constants

1) (a) The error on the training set will be smaller for both cases, and Ridge will produce an even smaller training error than LASSO. This is because with a small lamda, when parameter is small enough, Ridge heads to zero and stops caring about penalty.

   (b) The error on the testing set will be larger for both cases, and Ridge will produce an even larger testing error than LASSO. This is because Ridge stops caring about penalty when parameter gets small enough, so the learned function tends to be more overfitting, and thus Ridge will produce a larger testing error.

   (c) LASSO will have more small coefficients in w. The reason is that when parameter is small enough, Ridge does not care about penalty, while LASSO still has a linear penalty.

   (d) Ridge will have more nonzero coefficients in w. The reason is pretty much the same with part c. When parameter is very small, Ridge does not care about penalizing the coefficients, but LASSO still has linear penalty, so Ridge will have more nonzero coefficients in w.

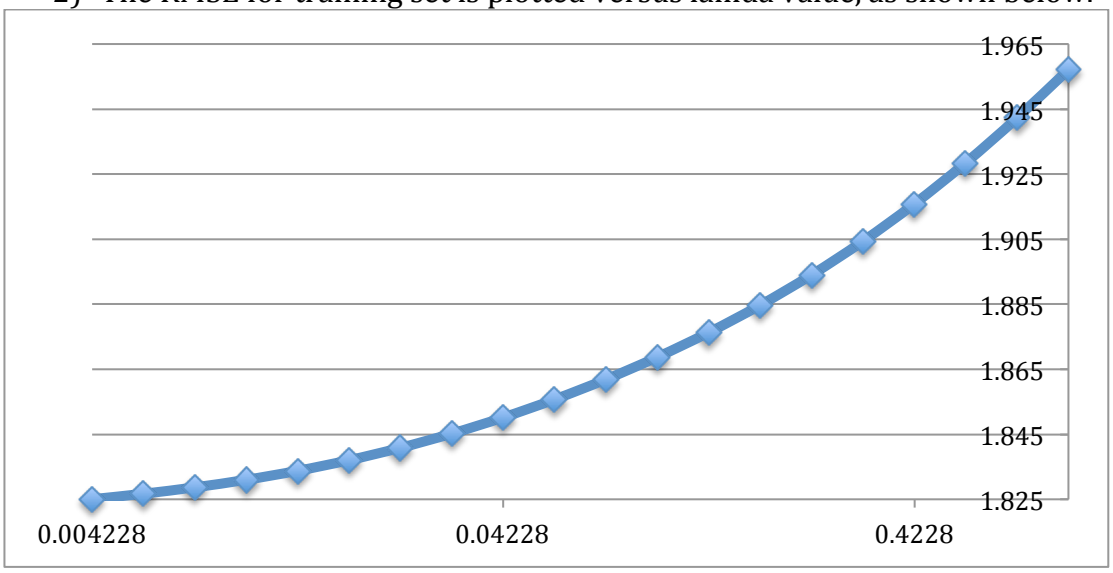
2) When lamda is too large (but not infinite), the difference between Ridge and LASSO gets smaller, but the difference still exists. This is because when lamda is large enough, it dominates the penalty term, and drives many coefficients towards zero. However, because Ridge increases quadratically, while LASSO increases linearly, Ridge will have an even

larger penalty than LASSO. Therefore, the training error of Ridge is larger than LASSO, and the test error should be large in both cases (but LASSO possibly has a larger test error), and Ridge will have more small coefficients in w, and LASSO will tend to have more nonzero coefficients.

4. Regression, Regularization, and Cross-Validation

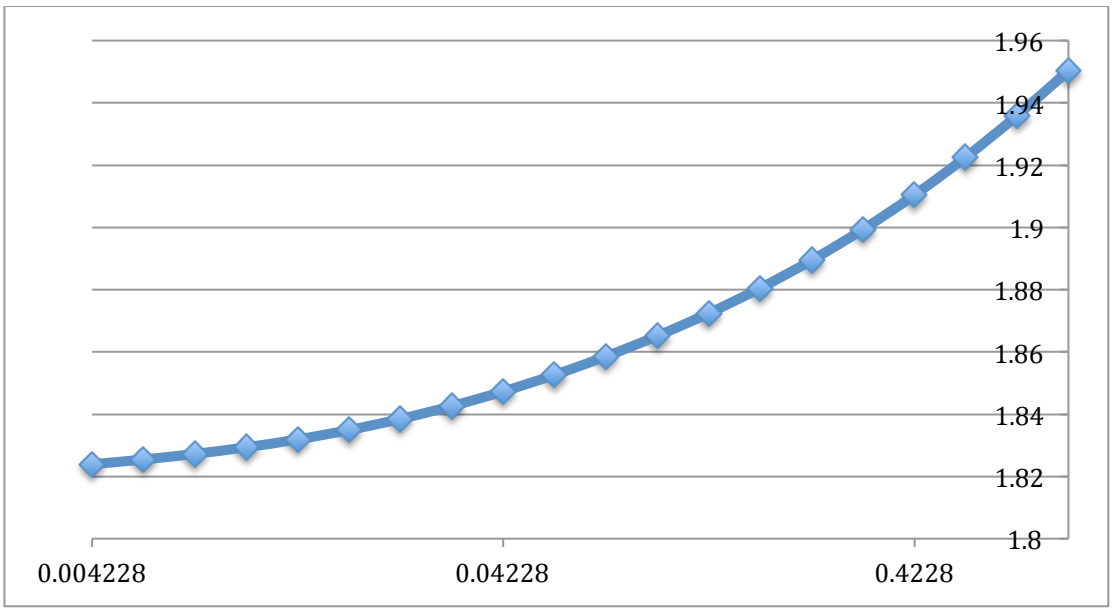
1) Training error is 1.80974735996
Test error is 2.36176469556

2) The RMSE for training set is plotted versus lamda value, as shown below.



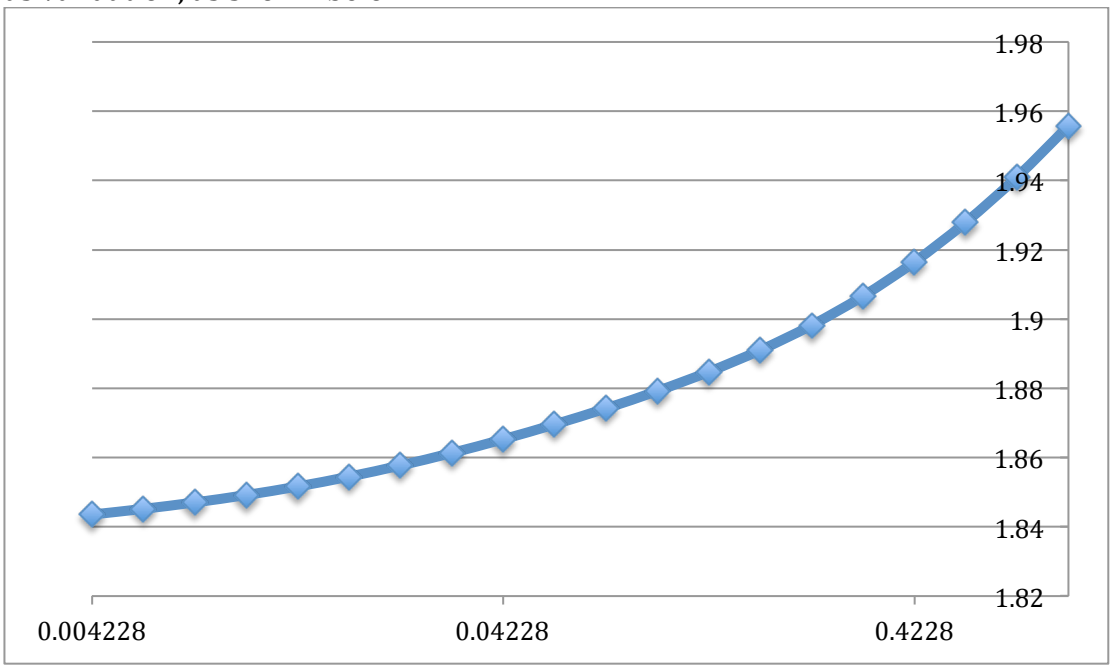
Although not indicated in graph, the lamda should have an optimal value when the cross validation has its minimum error, which is 1.870624694. In this case, lamda is 0.007516947, and the test case RSME is calculated to be 2.346090042.

3) The RMSE for training set is plotted versus lamda value with 10-fold, as shown below.



Although again not shown in graph, the cross validation error has its minimum value of 1.821180762, and the estimated optimal lamda is 0.042235136 in this case. The RMSE for test case with this lamda is calculated to be 2.349494479.

4) The RMSE for training set is plotted versus lamda value with rows 4000-5000 as validation, as shown below.



The validation error has its lowest value of 1.573529373, and the corresponding lamda value is 0.421875, which is likely to be the optimal lamda value. The RMSE for test case with this lamda is calculated to be 2.357453343. Compared to the k-fold cross validation method, this validation set method has much smaller validation errors, this might be because there is only one validation set, and the validation error is much more likely to be biased.