Final Report, Classification with the Census Income Data Set

My Dinh

12/8/2017

### I. Introduction:

This report focuses on the classification process for adult census data from 1994. We are looking to predict whether or not a person's salary is greater or less than 50K, based off of other attributes. The purpose of this project is to construct and analyze various models generated by classification trees.

For this report, we examine and clean the various features of our original data, and then selected the features that appeared to have the most influence on income prediction. We addressed the presence of class imbalance through different sampling techniques, and also tuned parameters with cross-validation before finalizing our models.

After performing feature engineering, we model three different classification trees and analyze the results. We find that our most successful tree that we built was our final RandomForest model, which ultimately had AUC value of 0.9125.

### II. Data:

The data used for this project came from the UCI repository, donated by Ronny Kohavi and Barry Becker. The basis of this data originated from the 1994 Census database. Multivariate data was collected onadult individuals, and was divided into a training set of 32561 observations and a test set of 16281 observations. There are 15 variables total, 6 of which are numerical and the other 9 variables are categorical.

Although most of our integer attributes were continuous, we noticed that education-num appeared to operate like a categorical variable, and had 16 levels. Our categorical attributes, with their relative number of levels, are: workclass(6), education(16), marital-status(7), occupation (14), relationship (6), race(5), sex(2), and native-country(41).

Income was also factored into two levels: <= 50K, and > 50K. 23.93% of the data had income >50K. After observations with missing values were removed, this probability became 24.78%.

At first glance, the original dataset was immense; creating an interpretable classification tree from this would need thoughtful data preparation. As such, we focused heavily on properly analyzing the data and examining each variable.

### III. Data Processing/Feature Engineering:

#### Data Cleaning

There is a total of 4262 missing values, for 2399 observations. Missing values are seen in the original dataset as '?' symbols. All the missing values belonged to three columns: occupation, nativecountry, and workclass, and multiple missing values were frequent for the same observation. As thus, it is very likely that the missing values are not at random (MNAR).

Given the MNAR, we examined whether or not a dataset with the missing values removed had better results than a dataset with imputed values. After

We will try to impute the missing values and remove it from the data and see what version works better.

### IV. Model Building:

#### A. Decision Tree

For our decision tree, we utilized functions from the tree and rpart library to fit our classification tree. In our baseline model based solely on the validation set, we found that only four variables (relationship, capital-gain, Edu\_Mean\_inc, and occupation) were used in the construction. There were 8 terminal nodes, and the misclassification rate was 15.8%.

To better fit this tree, we ran cross-validation to tune the tree complexity. With 10 folds, we found that the lowest cv error rate was 4120 cv errors, elbowing at 5 nodes. Deviation appeared to increase fairly linearly with k

As such, we fit our classification tree by this size (pruning by different levels of k did not significantly improve our accuracy rate, so we did not change that parameter), and found that our model to be the same as the baseline.

To confirm these results, we also pruned through the rpart library, using the cp values. We found the cp of the smallest tree, that was within 1 standard deviation of the tree with the smallest xerror. Our best xerror was 0.65678 with xstd of 0.0093883, so we selected the smallest tree with an error less than 0.6661683 This is the 4th tree, with cp = 0.010. Our test error is 0.006166048.

From this process, we again found that the resulting decision tree was equivalent to our baseline. The classification accuracy rate is 84.57%. The 5 most significant variables and their statistics were relationship (1861.25609), Gen\_Med\_Mrg\_Inc (1831.20301), marital\_status (1831.20301), Edu\_Mean\_inc (753.37400), and capital\_gain (762.63259).

The graph shows the variable importance as a proportion for each feature.

Our selected decision tree model had an AUC value of 0.8503 when we ran an ROC curve against the training validation set.

#### B. Bagged Trees

For bagging decision trees, the tuning parameter we examined is the number of trees to include. We did this by increasing the number of trees by 50 until we saw accuracy stopped showing improvement, and saw that accuracy leveled out at 200 trees. We also considered tuning to the depth of the tree (examining node size or the maximum number of nodes), but did not find a consistent parameter to utilize.

We then decided to account for class imbalance through different sampling procedures to see which one was the best. The model that we saw with the highest AUC value was the lower sampling technique with the ROC measure, with a training accuracy rate of 82.75%, TPR of 0.8316, specificity of 0.8263, and a FPR of 17.3746%

The top 6 most important features for this model, and their corresponding values, were relationship (100.000), fnlwgt (76.5515), capital-gain (31.8887), Edu\_Mean\_inc (27.7286), hours-per-week (27.3747), occupation (25.0592), age(19.3836), and occ\_sex (13.9597).

For our final bagged tree model, we utilized only the top 10 most important variables from the previous model. As such, our final model's ROC curve on the validation set had an AUC value 0.911.

#### c. Random Forest

#### V. Results and Conclusion:

When we ran our final decision tree model against the test dataset, we found that our ROC curve had a AUC value of 0.845. Our sensitivity (0.5057), specificity (0.9493), and FPR all remained fairly similar to what they were when we tested this model in our training validation set. The confusion matrix we obtained against the tet data is as follows:

Prediction Less.50k More.50k Less.50k 11805 1901 More.50k 630 1945

We found that our final bagged tree model had higher success in classification of income than our decision tree. The AUC against the test set was 0.897, which was slightly lower than our AUC for the validation set (0.911). Similarly, we saw that our sensitivity(0.8190), specificity(0.8051), and false positive rates (0.1949) all decreased slightly from what they were for the validation set. The model correctly classified 80.84% of the predictions; our confusion matrix is as follows:

Prediction Less.50k More.50k Less.50k 10011 696 More.50k 2424 3150