Final Report, Classification with the Census Income Data Set

My Dinh, Emily Liang

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).

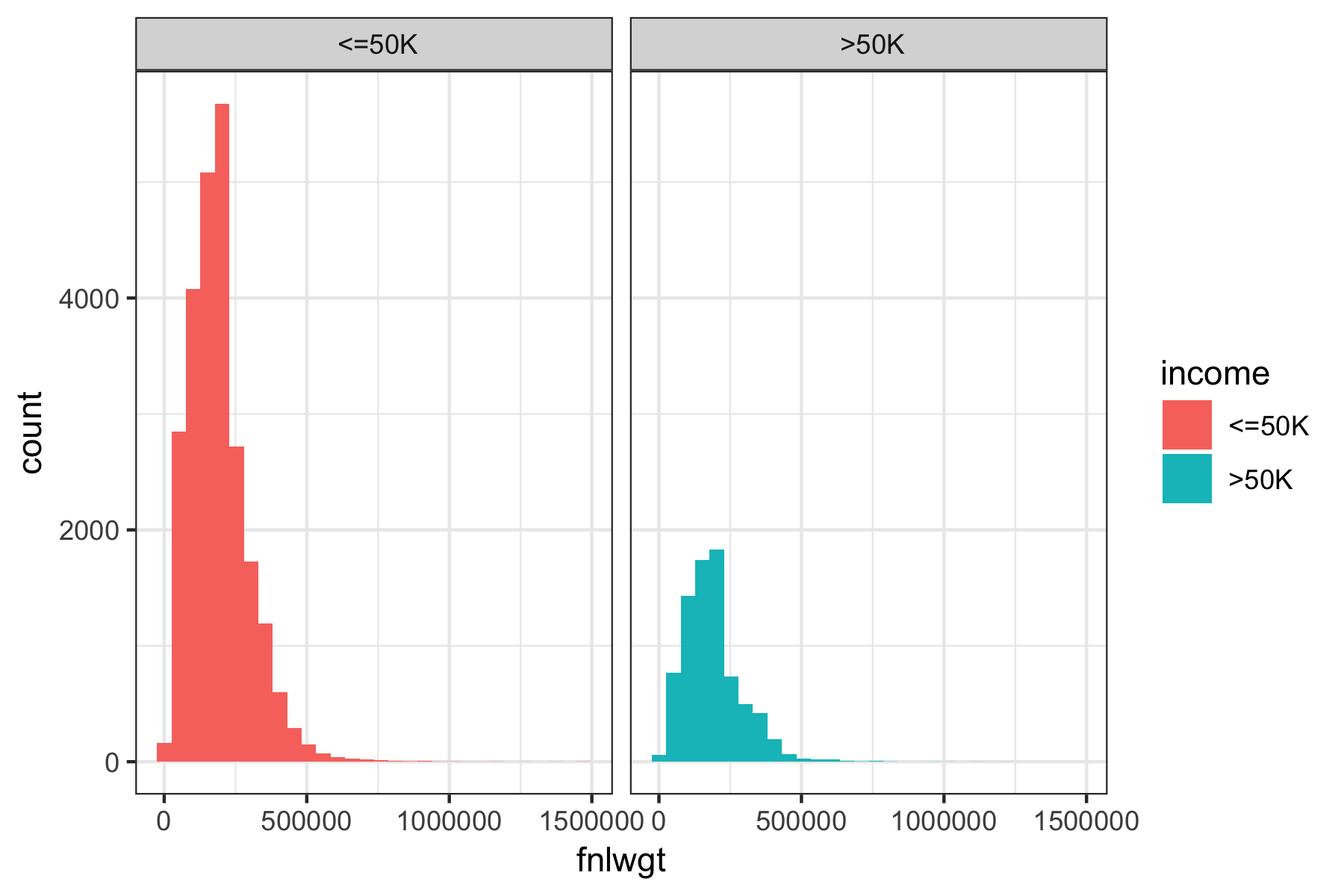
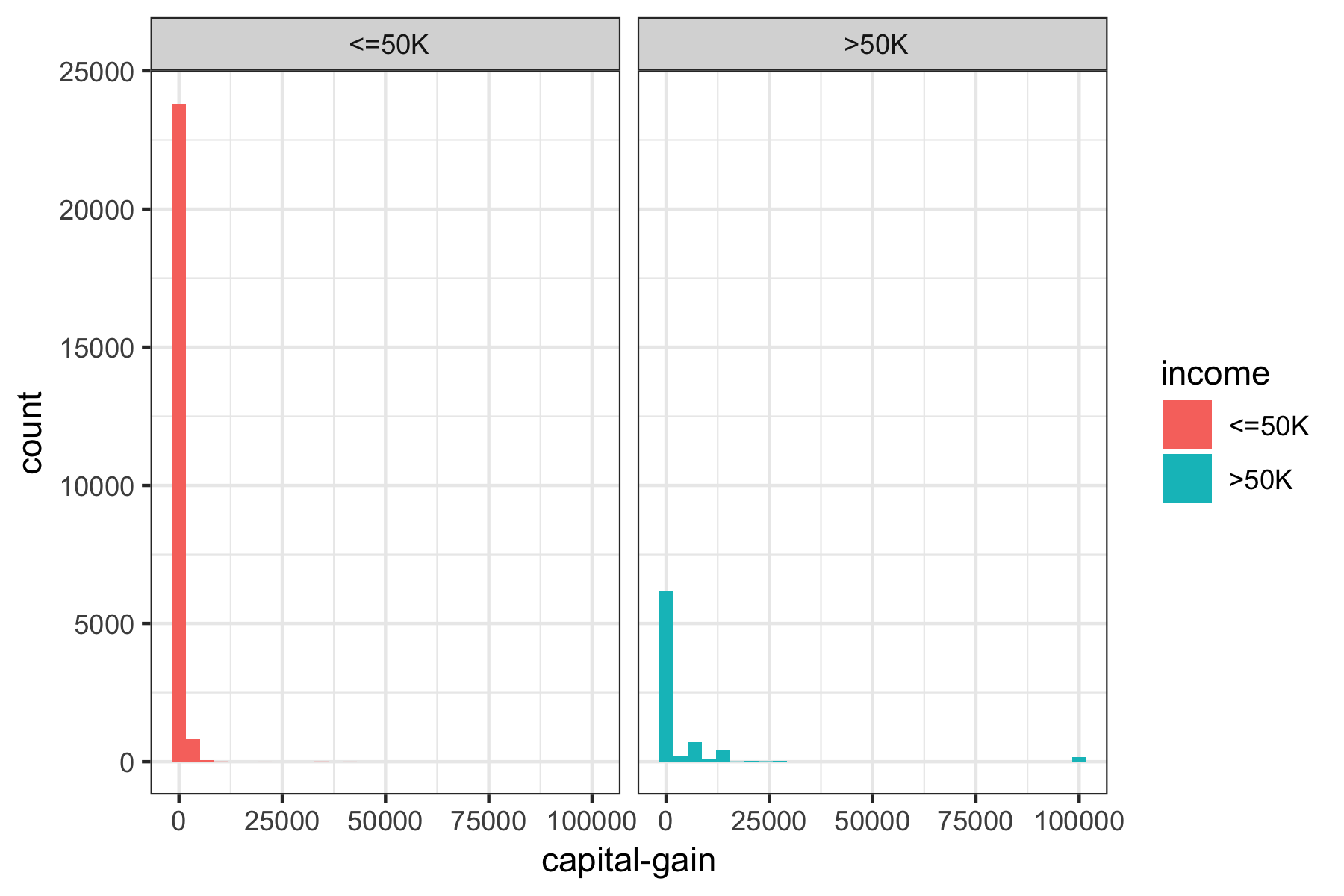
Income was also factored into two levels: <= 50K, and > 50K. 23.93% of the data had income >50K. After observations with missing values were removed, this probability became 24.78%.

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.

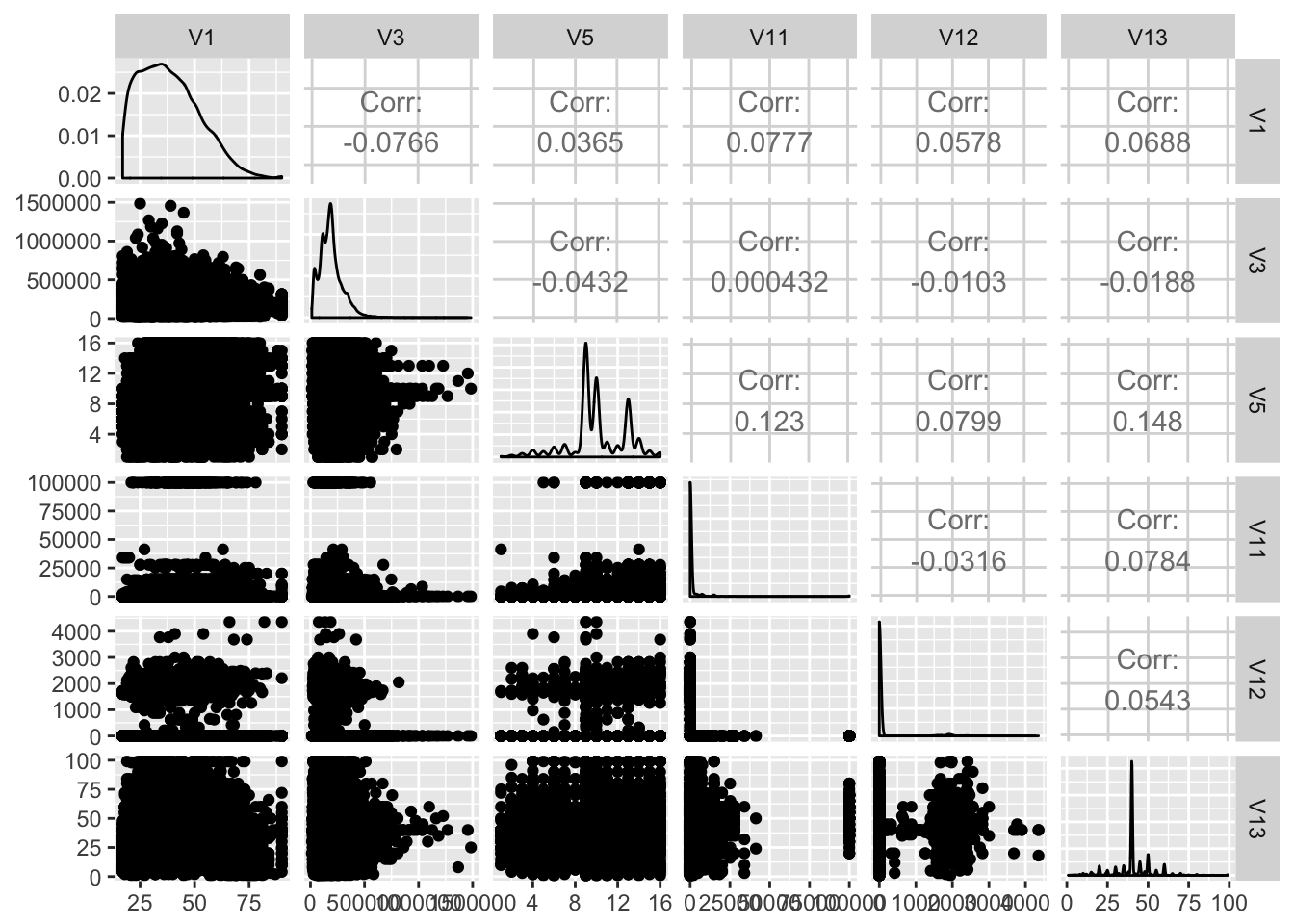
### III. Data Processing/Feature Engineering:

#### Data Cleaning/ Explonatory Analysis:

We first look for a relationship between income and other continous variables:

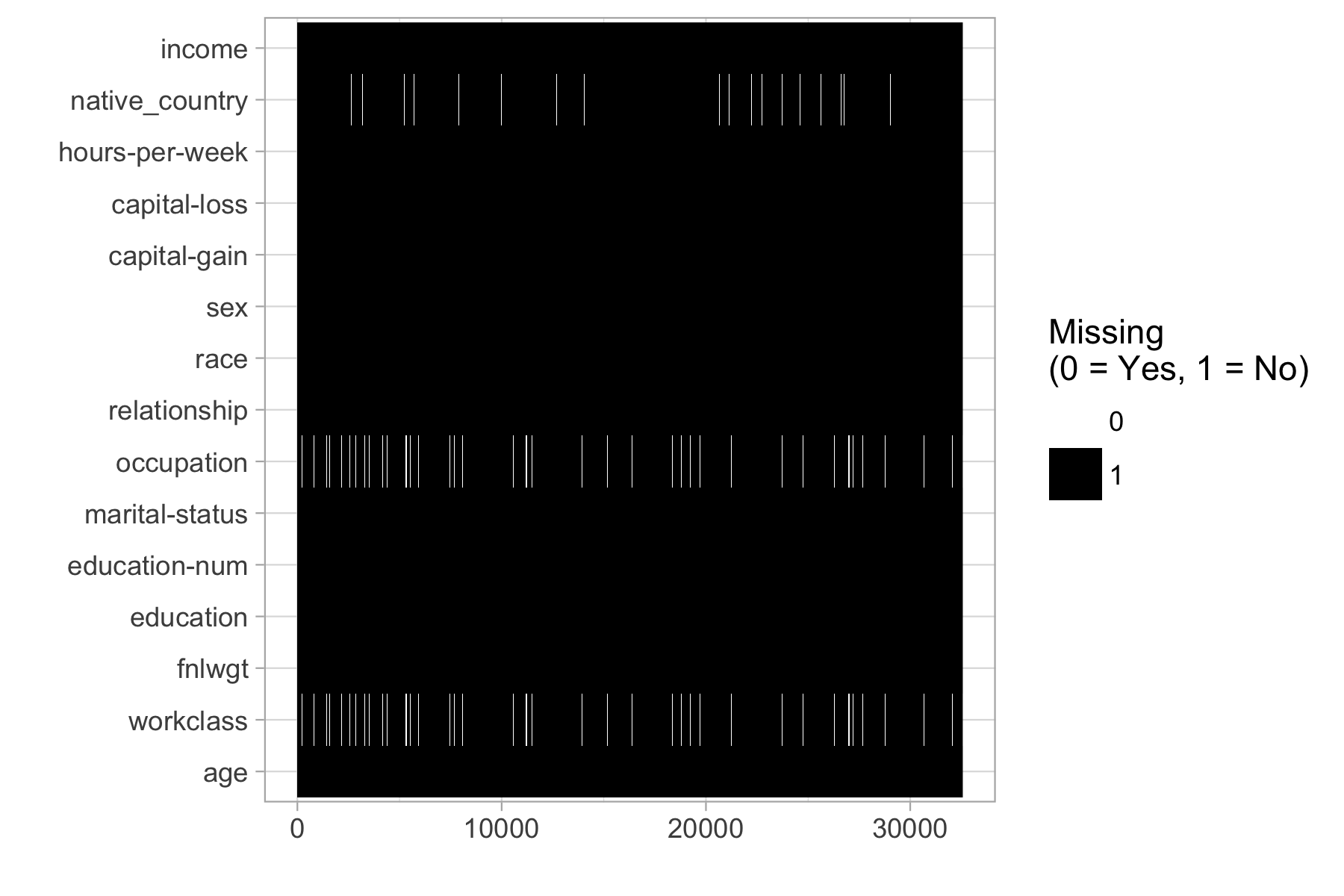
 \* Capital loss: people with income over 50K tend to have higher capital loss overall, but with smaller maximum capital loss. \* Capital gain: people with income over 50K tend to have higher capital gains, with some extreme outliers with values over 100k. We may need to remove them during the cleaning process. \* Age: the age distribution of people whose income less than 50K tends to skewed to the left, indicating that they are of a younger age. Meanwhile, the age distribution of people with income higher than 50K is roughly normal, with higher mean and median. \* Education number: people with higher income tend to have higher education-numbers \* fnlwgt: both income groups roughly share the same distribution for the final weight

We also look at the distribution of each continous variables and see if any relationship in the dataset. There isn’t any highly correlated pair of variables so there is no need to remove any variable at this stage.



*Missing values:*

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 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.

#### Feature Engineering:

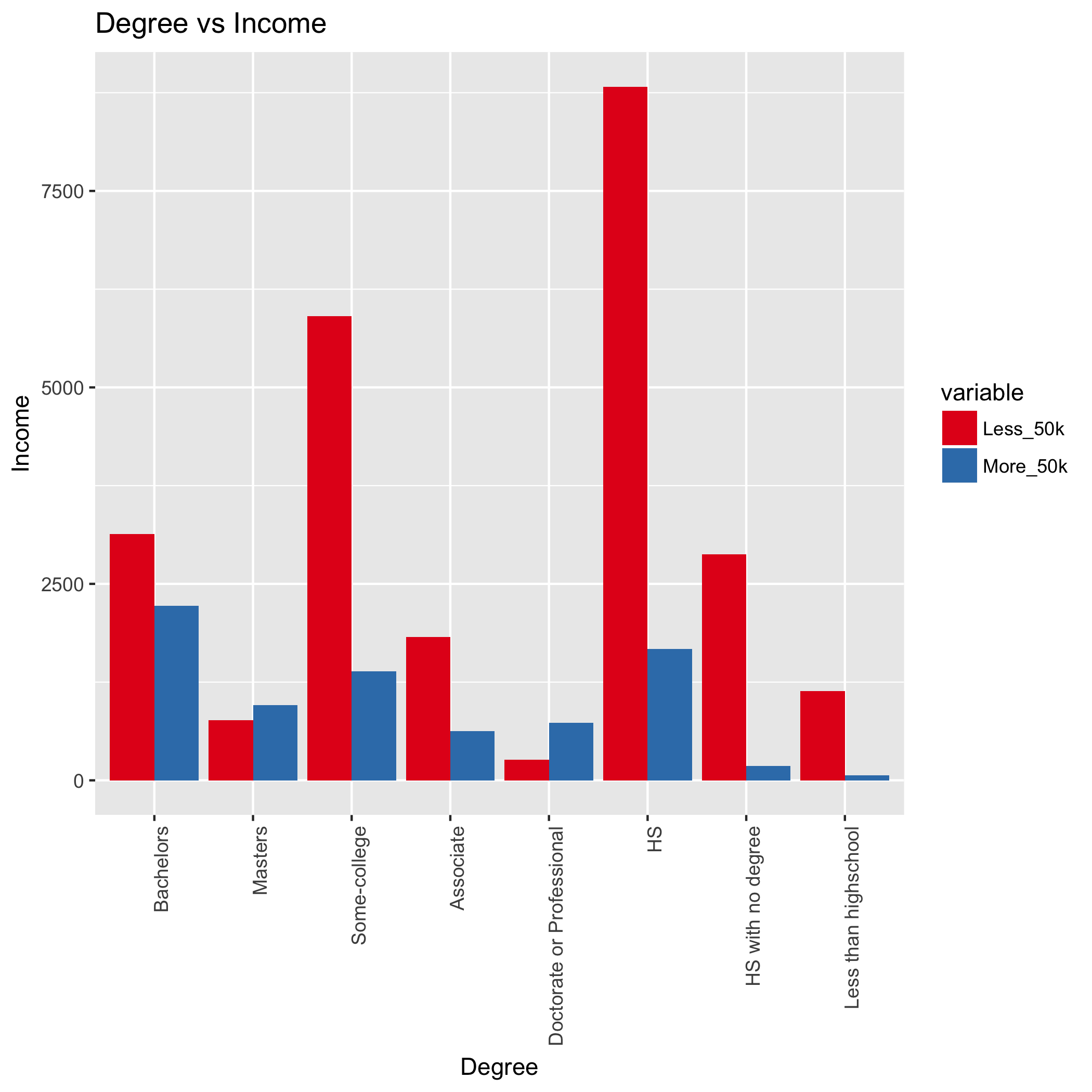
Since some categorical variables have more than 10 levels, this will affect the performance of tree based method for deciding splitting nodes, we reduce the number of levels by grouping based on our observation. We also refer to the income report for 1994 from census bureau. <https://www.census.gov/prod/2/pop/p60/p60-193.pdf>

New features are generated from the following variables: education, martial status, race, sex, and relationship.

1. Education:

We first combine educations levels into the following categories:

* Less than high school
* High school with no degree
* High school diploma or equivalent
* Some college, no degree
* Associate’s degree
* Bachelor’s degree
* Master’s degree
* Doctoral degree and professional degree



We observe that the income does seem to change with the degree, thus we decide to use the median income for each degree level that we found in the income report instead of the original education level. This new column is named Edu\_Mean\_Inc. We also removed education-num because it has the same information as the education level.

1. Age:

We decide to break age into 7 bins of ranges, to improve classification:

* 15- 24 years
* 25- 34 years
* 35 - 44 years
* 45 - 54 years
* 55 - 64 years
* 65 -74 years
* 75 years and older

We also replace each age category with median income from the income report.

1. Race:

We first combined “Amer-Indian-Eskimo” with “Other” since the ratio is small. We observed that less females of minorities class earn more than 50K than white female. We decided to make a new column based on gender and minority status. For example, if individual is female and is black, this person will in class “fb”. We also replace the actual catergory with median income that associate with the specified gender and race. The new column is gen\_race.

Race variable is replaced with median income that associated with race in the new variable Race\_Med\_Inc.

1. Marital\_status

* For marital status, we split the groups into based on whether or not that person was married, considered to legally have a spouse, and whether or not their spouse was living with them. “Has-Spouse” indicates that the person was married and with their spouse (Married-civ-spouse for civilian spouses, and Married-AF-spouse for spouses from the armed forces), whereas “Married-spouse-absent” and “Separated” denoted an “Absent-Spouse”. Divorced and Widowed were placed under the same category as “No-Spouse”, while Never-married stayed the same but was renamed “Single”.
* We observed income level is affected by both gender and martial status. Thus we create a new column that combine both information of gender and martial status, then replace each category with the median income. For example, person who is has spouse absent and is female will have median income while male has spouse absent has income . This column is name Gen\_Med\_Mrg\_Inc.

1. Occupation:

* Splitting up Occupation was largely based on three levels - white vs blue collar, service vs not service, and then skilled vs. not skilled for not service jobs.
* Similar to what we did with gender and martial status, we also combine information of occupation and gender into a new variable occ\_sex and replace the level with median income.
* Because our new columns contain information about gender, we decided to remove gender column.

1. Workclass:

The workclass variable currently has 8 different levels - Federal-gov, Local-goc, State-gov, Private, Self-emp-inc, Self-emp-not-inc, Without pay, and Never-worked. We decided to group all the government jobs together, after examining the income ratio for each. We did not combine the self employed classes together, given that self employed incorporated positions tend to have higher pay than not-incorporated ones. Finally, We combined those that never worked with those that were without pay into a new group, of those who were not working.

1. Relationships:

For relationships, this is the relationship of the surveyor to a central household figure. Unmarried referred to an unmarried partner; as such, we combined that we “Not-in-family” to make a grouping of Not-relative relationships. Husband and Wife were grouped together under “Spouse”, while the last two remaining classes were combined as “Other-relative” relationships.

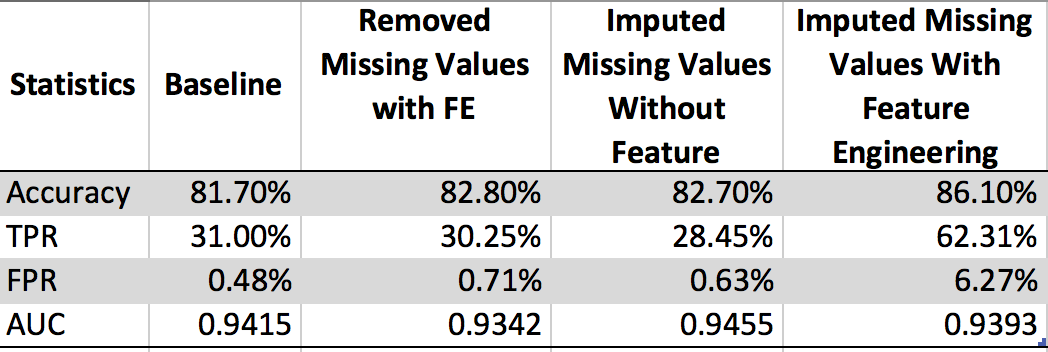
1. Capital\_gain:

We noticed that there appear to be outliers for capital gain (there was a chunk of data that had values of over 99999). As such, we decided to change these outlier values to the mean value of the capital gains.

### IV. Model Building:

Because of imbalanced class situation, instead of randomly sample data to divide into train and test set, we used stratified sampling to make sure that the ratio between 2 classes are the same in each dataset. We use 80% of data for the train set and 20% for the validation set.

We original build a simple model on four datasets, the dataset with missing value removed with feature engineering and with original features, and the dataset with missing values imputed by misForest with feature engineering and with original features. This simple model includes a decision tree, bagges trees with 100 tress, and random forest with 100 trees without any hyperparameter tunning. The baseline models are defined as the models on the original dataset without any feature engineering and with missing values removed. The models’ performance using random forest is reported below:



Since the imputed data set with feature engineering have significant higher positive rate and accuracy, we decide to choose the imputed missing values with feature engineering as our final dataset to build our models.

Except from decision trees, we fit models using indepedent categories (fit model using formula income~., data) and grouped categories (fit model using format xtrain, ytrain). With indepedent categories, model make a dummify matrix for each categories variable,thus the feature importance score isn’t very expressive. Meanwhile, running models with group categories increase model complexity. However, with our sampling methods described below, some models can’t be computed because of dimmension limitted. The models wih independent categories perform worse than grouped categories ones, thus we decided to use xtrain, ytrain format to fit our models.

We noticed that even with our best model, it still does not do a good job with predicting people with income more than 50k, which is caused by the imbalance class problems. Thus we try different techniques for bagged trees and random forest models. We build three types of models:

##### 1. Sampling methods with accuracy metric:

The problem with imblanced class is that the event rate of minority class is significant less than the minority class. One natural way to fix this issue is sampling each class to balance the event rates. We employ four sampling methods:

* Up sampling: sample with replacement from the minority class until each class has the approximately the same number.
* Down sampling:select data points from the majority class so that the majority class is roughly the same size as the minority class.
* Down sampling with bootstrap sampling: create a bootstrap sample roughly a same size from each class. The choice of sample here is 50.
* SMOTE sampling: use both up-sampling and down sampling. While down-sampling in smote is the same as described aboe, to up-sampling the minority class, SMOTE synthesizes new cases by sampling from the minority class and its KNN.

The models are built using accuracy as the metric, in other word, the model fit the parameter to maximize accuracy.

##### 2. Sampling methods with AUC metric:

We use the same sampling methods that were described in the previous sections but maximizing AUC to build our models. By using AUC metrics, we hope to improve the performance for minority class instead of only targetting majority class.

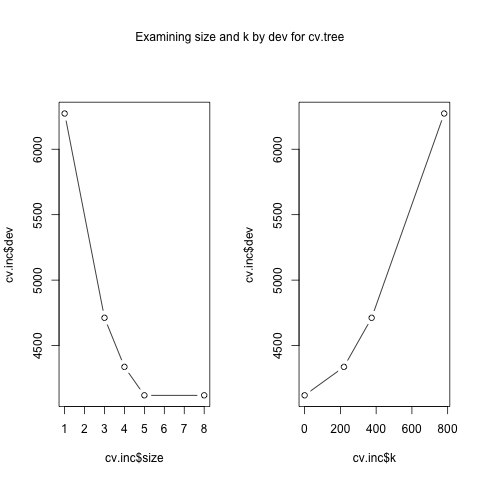
##### 3. Class weights and sampling methods with AUC metric:

Beside from sampling methods and AUC metrics, we also use unequal case weights to penalize the majority class. The class weights we implements for class i is computed as: $\frac{1}{#classes \* #observation for class i}$

#### A. Decision Tree

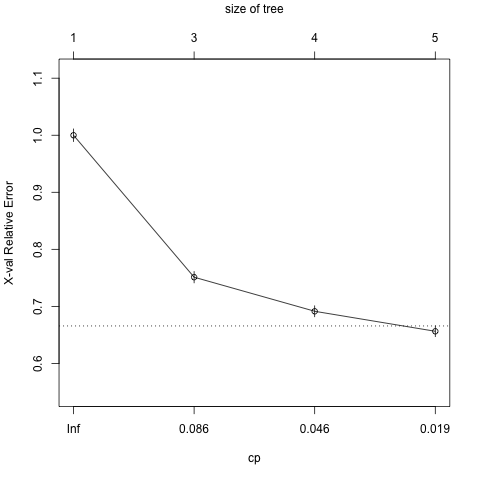
For our decision tree, we utilized functions from the tree and rpart library to fit our classification tree. In our baseline model based solely on the validation set, we found that only four variables (relationship, capital-gain, Edu\_Mean\_inc, and occupation) were used in the construction. There were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

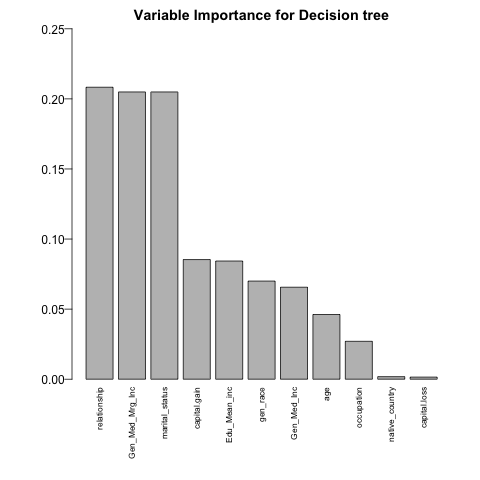


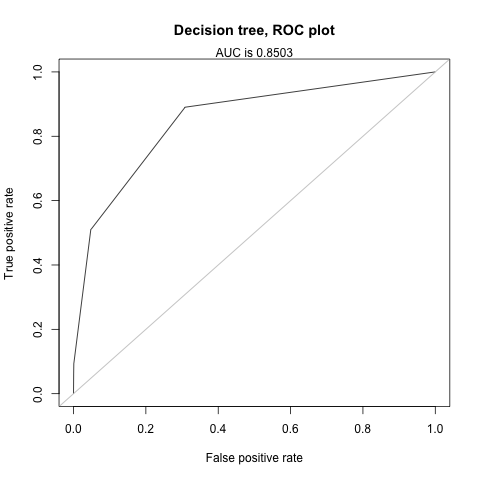
As such, we fit our classification tree by this size (pruning by different levels of k did not significantly improve our accuracy rate, so we did not change that parameter), and found that our model to be the same as the baseline.

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).

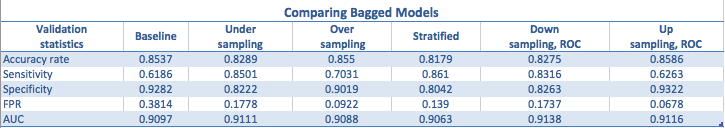
The graph shows the variable importance as a proportion for each feature. 

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 down sampling methods 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 other models and their various statistics when modeled against the test validation set are shown in the table below.



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).

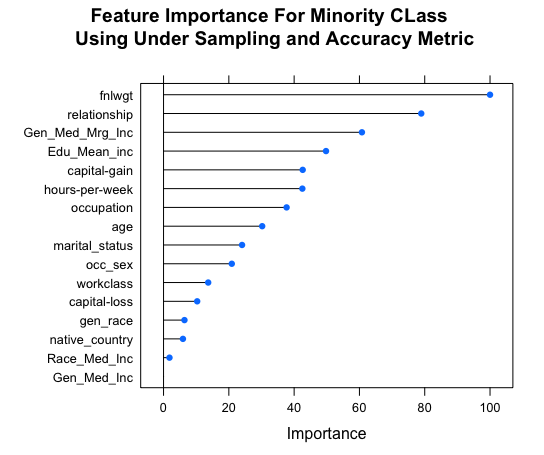
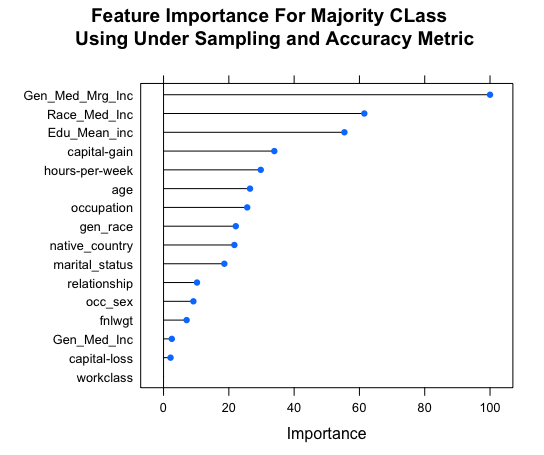
For our final bagged tree model, we utilized only the top 10 most important variables from the previous model. As such, our final model’s ROC curve on the validation set had an AUC value 0.911.

#### C. Random Forest

We fit a random forest with 10 from caret package using three techniques as described above. Random search and grid search with 5-fold cross validation are implemented to tune hyperparameter simultaneoulsy while training the model to find which model that perform the best. Two tunning parameters are mtries and ntree.

We observe that the TPR increases signficantly from 10-20% compared to the baseline models. Accurary and AUC also increases with new models. Out of 12 models, the model using bootstrap stratified down sampling does not perform that well compared to other models.

Since each model performs differently in each class, we look at the feature importance scores for the model that perform well for the majority class (Less than 50k) and the model that perform well for the minority class.

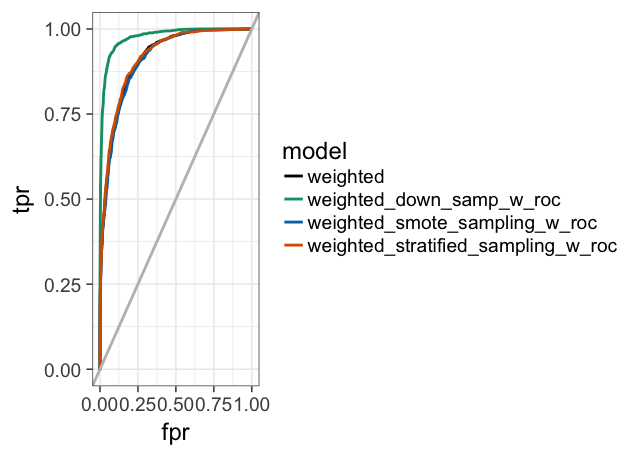


The plots indicate that while fnlwgt, relationship don’t contribute well in model for majority class, it is the most importance feature to predict for minority class. The most common featues for both classes are: Gen\_Med\_Mrg\_Inc, Edu\_Mean\_Inc, captial\_gain, hours\_per\_week, age.

To select the best model, we base on 4 statistics: accurary, TPR, FPR, and AUC. Our best random forest model is the model using down sampling method with unequal class weights using ROC as metrics. From this model, we look at importance features to select the top 10 features and retrain the model with top ten features to predict the test set. The tuning parameter for this model: ntree = 100, mtry = 3. Below is the importance score for the top 5 features and the statistics of chosen models:

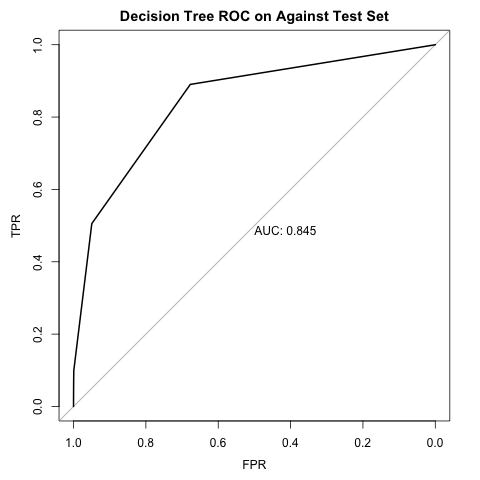
|  |  |
| --- | --- |
| Features | Scores (scaled) |
| fnlwgt | 100.000 |
| Gen\_Med\_Mrg\_Inc | 93.975 |
| relationship | 92.981 |
| Edu\_Mean\_inc | 83.541 |
| occupation | 69.536 |

|  |  |
| --- | --- |
| Statistics | Rate |
| Accuracy | 89.63% |
| TPR | 94.52% |
| FPR | 11.91% |
| TNR | 88.09% |
| AUC | 97.71% |

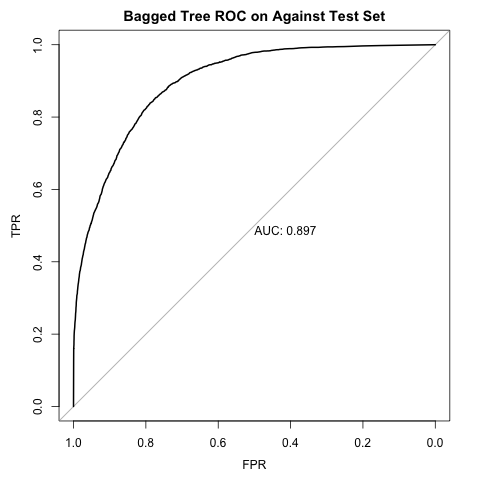


#### V. Results and Conclusion:

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 (0.0507) all remained fairly similar to what they were when we tested this model in our training validation set.



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 final random forest models perform better than the decision and bagged trees. Our confusion matrix for this model is reported at:

|  |  |  |
| --- | --- | --- |
| Prediction | Less.50k | More.50k |
| Less.50k | 10032 | 636 |
| More.50k | 2403 | 3210 |

Our accuracy rate is at 81.3% with TPR of 83.46% and AUC of 0.9137.

